



Polisis: Automated Analysis and Presentation of Privacy Policies Using Deep Learning

*Hamza Harkous, École Polytechnique Fédérale de Lausanne (EPFL);
Kassem Fawaz, University of Wisconsin-Madison; Rémi Lebret, École Polytechnique Fédérale
de Lausanne (EPFL); Florian Schaub and Kang G. Shin, University of Michigan;
Karl Aberer, École Polytechnique Fédérale de Lausanne (EPFL)*

<https://www.usenix.org/conference/usenixsecurity18/presentation/harkous>

**This paper is included in the Proceedings of the
27th USENIX Security Symposium.**

August 15–17, 2018 • Baltimore, MD, USA

ISBN 978-1-931971-46-1

**Open access to the Proceedings of the
27th USENIX Security Symposium
is sponsored by USENIX.**

Polisis: Automated Analysis and Presentation of Privacy Policies Using Deep Learning

Hamza Harkous¹, Kassem Fawaz², Rémi Lebret¹, Florian Schaub³, Kang G. Shin³, and Karl Aberer¹

¹ École Polytechnique Fédérale de Lausanne (EPFL)

² University of Wisconsin-Madison

³ University of Michigan

Abstract

Privacy policies are the primary channel through which companies inform users about their data collection and sharing practices. These policies are often long and difficult to comprehend. Short notices based on information extracted from privacy policies have been shown to be useful but face a significant *scalability* hurdle, given the number of policies and their evolution over time. Companies, users, researchers, and regulators still lack usable and scalable tools to cope with the breadth and depth of privacy policies. To address these hurdles, we propose an automated framework for privacy **policy analysis (Polisis)**. It enables scalable, dynamic, and multi-dimensional queries on natural language privacy policies. At the core of Polisis is a privacy-centric language model, built with 130K privacy policies, and a novel hierarchy of neural-network classifiers that accounts for both high-level aspects and fine-grained details of privacy practices. We demonstrate Polisis' modularity and utility with two applications supporting *structured* and *free-form* querying. The structured querying application is the automated assignment of privacy icons from privacy policies. With Polisis, we can achieve an accuracy of 88.4% on this task. The second application, PriBot, is the first free-form question-answering system for privacy policies. We show that PriBot can produce a correct answer among its top-3 results for 82% of the test questions. Using an MTurk user study with 700 participants, we show that at least one of PriBot's top-3 answers is relevant to users for 89% of the test questions.

1 Introduction

Privacy policies are one of the most common ways of providing notice and choice online. They aim to inform users how companies collect, store and manage their personal information. Although some service providers have improved the comprehensibility and readability of their privacy policies, these policies remain excessively long and difficult to follow [1, 2, 3, 4, 5]. In 2008, Mc-

Donald and Cranor [4] estimated that it would take an average user 201 hours to read all the privacy policies encountered in a year. Since then, we have witnessed a smartphone revolution and the rise of the Internet of Things (IoTs), which lead to the proliferation of services and associated policies [6]. In addition, emerging technologies brought along new forms of user interfaces (UIs), such as voice-controlled devices or wearables, for which existing techniques for presenting privacy policies are not suitable [3, 6, 7, 8].

Problem Description. Users, researchers, and regulators are not well-equipped to process or understand the content of privacy policies, especially at scale. Users are surprised by data practices that do not meet their expectations [9], hidden in long, vague, and ambiguous policies. Researchers employ expert annotators to analyze and reason about a subset of the available privacy policies [10, 11]. Regulators, such as the U.S. Department of Commerce, rely on companies to self-certify their compliance with privacy practices (e.g., the Privacy Shield Framework [12]). The *problem* lies in stakeholders lacking the usable and scalable tools to deal with the breadth and depth of privacy policies.

Several proposals have aimed at alternative methods and UIs for presenting privacy notices [8], including machine-readable formats [13], nutrition labels [14], privacy icons (recently recommended by the EU [15]), and short notices [16]. Unfortunately, these approaches have faced a significant *scalability* hurdle: the human effort needed to retrofit the new notices to existing policies and maintain them over time is tremendous. The existing research towards automating this process has been limited in scope to a handful of "queries," e.g., whether the policy mentions data encryption or whether it provides an opt-out choice from third-party tracking [16, 17].

Our Framework. We overcome this scalability hurdle by proposing an automatic and comprehensive framework for privacy **policy analysis (Polisis)**. It divides a privacy policy into smaller and self-contained fragments

of text, referred to as *segments*. Polisis automatically annotates, with high accuracy, each segment with a set of labels describing its data practices. Unlike prior research in automatic labeling/analysis of privacy policies, Polisis does not just predict a handful of classes given the entire policy document. Instead, Polisis annotates the privacy policy at a much finer-grained scale. It predicts for each segment the set of classes that account for both the high-level aspects and the fine-grained classes of embedded privacy information. Polisis uses these classes to enable scalable, dynamic, and multi-dimensional queries on privacy policies, in a way not possible with prior approaches.

At the core of Polisis is a novel hierarchy of neural-network classifiers that involve 10 high-level and 122 fine-grained privacy classes for privacy-policy segments. To build these fine-grained classifiers, we leverage techniques such as subword embeddings and multi-label classification. We further seed these classifiers with a custom, privacy-specific language model that we generated using our corpus of more than 130,000 privacy policies from websites and mobile apps.

Polisis provides the underlying intelligence for researchers and regulators to focus their efforts on merely designing a set of queries that power their applications. We stress, however, that Polisis is not intended to replace the privacy policy – as a legal document – with an automated interpretation. Similar to existing approaches on privacy policies’ analysis and presentation, it decouples the legally binding functionality of these policies from their informational utility.

Applications. We demonstrate and evaluate the modularity and utility of Polisis with two robust applications that support *structured* and *free-form* querying of privacy policies.

The *structured querying* application involves extracting short notices in the form of privacy icons from privacy policies. As a case study, we investigate the Disconnect privacy icons [18]. By composing a set of simple rules on top of Polisis, we show a solution that can automatically select appropriate privacy icons from a privacy policy. We further study the practice of companies assigning icons to privacy policies at scale. We empirically demonstrate that existing privacy-compliance companies, such as TRUSTe (now rebranded as TrustArc), might be adopting permissive policies when assigning such privacy icons. Our findings are consistent with anecdotal controversies and manually investigated issues in privacy certification and compliance processes [19, 20, 21].

The second application illustrates the power of *free-form querying* in Polisis. We design, implement and evaluate PriBot, the first automated Question-Answering (QA) system for privacy policies. PriBot extracts the

relevant privacy policy segments to answer the user’s free-form questions. To build PriBot, we overcame the non-existence of a public, privacy-specific QA dataset by casting the problem as a ranking problem that could be solved using the classification results of Polisis. PriBot matches user questions with answers from a previously unseen privacy policy, in real time and with high accuracy – demonstrating a more intuitive and user-friendly way to present privacy notices and controls. We evaluate PriBot using a new test dataset, based on real-world questions that have been asked by consumers on Twitter.

Contributions. With this paper we make the following contributions:

- We design and implement Polisis, an approach for automatically annotating previously unseen privacy policies with high-level and fine-grained labels from a pre-specified taxonomy (Sec. 2, 3, 4, and 5).
- We demonstrate how Polisis can be used to assign privacy icons to a privacy policy with an average accuracy of 88.4%. This accuracy is computed by comparing icons assigned with Polisis’ automatic labels to icons assigned based on manual annotations by three legal experts from the OPP-115 dataset [11] (Sec. 6).
- We design, implement and evaluate PriBot, a QA system that answers free-form user questions from privacy policies (Sec. 7). Our accuracy evaluation shows that PriBot produces at least one correct answer (as indicated by privacy experts) in its top three for 82% of the test questions and as the top one for 68% of the test questions. Our evaluation of the perceived utility with 700 MTurk crowdworkers shows that users find a relevant answer in PriBot’s top-3 for 89% of the questions (Sec. 8).
- We make Polisis publicly available by providing three web services demonstrating our applications: a service giving a visual overview of the different aspects of each privacy policy, a chatbot for answering user questions in real time, and a privacy-labels interface for privacy policies. These services are available at <https://pribot.org>.

2 Framework Overview

Fig. 1 shows a high-level overview of Polisis. It comprises three layers: *Application Layer*, *Data Layer*, and *Machine Learning (ML) Layer*. Polisis treats a privacy policy as a list of semantically coherent segments (i.e., groups of consecutive sentences). It also utilizes a taxonomy of privacy data practices. One example of such a taxonomy was introduced by Wilson *et al.* [11] (see also Fig. 3 in Sec. 4).

Application Layer (Sec. 5, 6 & 7): The Application Layer provides fine-grained information about the

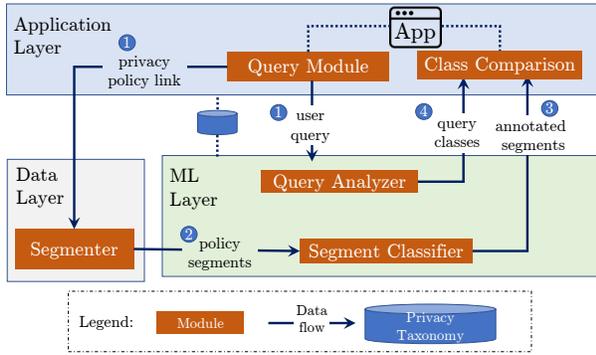


Fig. 1: A high-level overview of Polisis.

privacy policy, thus providing the users with high modularity in posing their queries. In this layer, a *Query Module* receives the *User Query* about a privacy policy (Step 1 in Fig. 1). These inputs are forwarded to lower layers, which then extract the privacy classes embedded within the query and the policy’s segments. To resolve the user query, the *Class-Comparison* module identifies the segments with privacy classes matching those of the query. Then, it passes the matched segments (with their predicted classes) back to the application.

Data Layer (Sec. 3): The Data Layer first scrapes the policy’s webpage. Then, it partitions the policy into semantically coherent and adequately sized segments (using the *Segmenter* component in Step 2 of Fig. 1). Each of the resulting segments can be independently consumed by both the humans and programming interfaces.

Machine Learning Layer (Sec. 4): In order to enable a multitude of applications to be built around Polisis, the ML layer is responsible for producing rich and fine-grained annotations of the data segments. This layer takes as an input the privacy-policy segments from the Data Layer (Step 2) and the user query (Step 1) from the Application Layer. The *Segment Classifier* probabilistically assigns each segment a set of class-value pairs describing its data practices. For example, an element in this set can be *information-type=location* with probability $p = 0.65$. Similarly, the *Query Analyzer* extracts the privacy classes from the user’s query. Finally, the class-value pairs of both the segments and the query are passed back to the *Class Comparison* module of the Application Layer (Steps 3 and 4).

3 Data Layer

To pre-process the privacy policy, the Data Layer employs a *Segmenter* module in three stages: extraction, list handling, and segmentation. The Data Layer requires no information other than the link to the privacy policy.

Policy Extraction: Given the URL of a privacy policy, the segmenter employs Google Chrome in headless mode (without UI) to scrape the policy’s web-

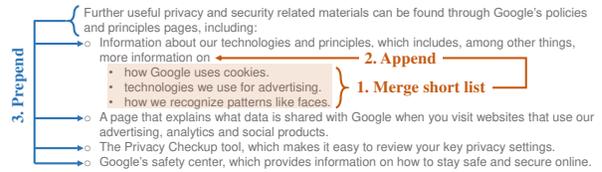


Fig. 2: List merging during the policy segmentation.

page. It waits for the page to fully load which happens after all the JavaScript has been downloaded and executed. Then, the segmenter removes all irrelevant HTML elements including the scripts, header, footer, side/navigation menus, comments, and CSS.

Although several online privacy policies contain dynamically viewable content (e.g., accordion toggles and collapsible/expandable paragraphs), the “dynamic” content is already part of the loaded webpage in almost all cases. For example, when the user expands a collapsible paragraph, a local JavaScript exposes an offline HTML snippet; no further downloading takes place.

We confirmed this with the privacy policies of the top 200 global websites from Alexa.com. For each privacy-policy link, we compared the segmenter’s scraped content to that extracted from our manual navigation of the same policy (while accounting for all the dynamically viewable elements of the webpage). Using a fuzzy string matching library,¹ we found that the segmenter’s scraped policy covers, on average, 99.08% of the content of the manually fetched policy.

List Aggregation: Second, the segmenter handles any ordered/unordered lists inside the policy. Lists require a special treatment since counting an entire lengthy list, possibly covering diverse data practices, as a single segment could result in noisy annotations. On the other hand, treating each list item as an independent segment is problematic as list elements are typically not self-contained, resulting in missed annotations. See Fig. 2 from Google’s privacy policy as an example².

Our handling of the lists involves two techniques: one for short list items (e.g., the inner list of Fig. 2) and another for longer list items (e.g., the outer list of Fig. 2). For short list items (maximum of 20 words per element), the segmenter combines the elements with the introductory statement of the list into a single paragraph element (with `<p>` tag). The rest of the lists with long items are transformed into a set of paragraphs. Each paragraph is a distinct list element prepended by the list’s introductory statement (Step 3 in Fig. 2).

¹<https://pypi.python.org/pypi/fuzzywuzzy>

²<https://www.google.com/intl/en-US/policies/privacy/archive/20160829/>, last modified on Aug. 29, 2016, retrieved on Jun. 27, 2018

Policy Segmentation: The segmenter performs an initial coarse segmentation by breaking down the policy according to the HTML `<div>` and `<p>` tags. The output of this step is an initial set of policy segments. As some of the resulting segments might still be long, we subdivide them further with another technique. We use GraphSeg [22], an unsupervised algorithm that generates semantically coherent segments. It relies on word embeddings to generate segments as cliques of related (semantically similar) sentences. For that purpose, we use custom, domain-specific word embeddings that we generated using our corpus of 130K privacy policies (*cf.* Sec. 4). Finally, the segmenter outputs a series of fine-grained segments to the Machine Learning Layer, where they are automatically analyzed.

4 Machine Learning Layer

This section describes the components of Polisis’ Machine Learning Layer in two stages: (1) an *unsupervised* stage, in which we build domain-specific word vectors (i.e., word embeddings) for privacy policies from unlabeled data, and (2) a *supervised stage*, in which we train a novel hierarchy of privacy-text classifiers, based on neural networks, that leverages the word vectors. These classifiers power the *Segment Classifier* and *Query Analyzer* modules of Fig. 1. We use word embeddings and neural networks thanks to their proven advantages in text classification [23] over traditional techniques.

4.1 Privacy-Specific Word Embeddings

Traditional text classifiers use the words and their frequencies as the building block for their features. They, however, have limited generalization power, especially when the training datasets are limited in size and scope. For example, replacing the word “erase” by the word “delete” can significantly change the classification result if “delete” was not in the classifier’s training set.

Word embeddings solve this issue by extracting generic word vectors from a large corpus, in an unsupervised manner, and enabling their use in new classification problems (a technique termed *Transfer Learning*). The features in the classifiers become the word vectors instead of the words themselves. Hence, two text segments composed of semantically similar words would be represented by two groups of word vectors (i.e., features) that are close in the vector space. This allows the text classifier to account for words outside the training set, as long as they are part of the large corpus used to train the word vectors.

While general-purpose pre-trained embeddings, such as Word2vec [24] and GloVe [25] do exist, domain-specific embeddings result in better classification accuracy [26]. Thus, we trained custom word embeddings for the privacy-policy domain. To that end, we created a corpus of 130K privacy policies collected from apps on

the Google Play Store. These policies typically describe the overall data practices of the apps’ companies.

We crawled the metadata of more than 1.4 million Android apps available via the PlayDrone project [27] to find the links to 199,186 privacy policies. We crawled the web pages for these policies, retrieving 130,326 policies which returned an HTTP status code of 200. Then, we extracted the textual content from their HTML using the policy crawler described in Sec. 3. We will refer to this corpus as the *Policies Corpus*. Using this corpus, we trained a word-embeddings model using *fastText* [28]. We henceforth call this model the *Policies Embeddings*. A major advantage of using *fastText* is that it allows training vectors for *subwords* (or character n -grams of sizes 3 to 6) in addition to words. Hence, even if we have words outside our corpus, we can assign them vectors by combining the vectors of their constituent subwords. This is very useful in accounting for spelling mistakes that occur in applications that involve free-form user queries.

4.2 Classification Dataset

Our Policies Embeddings provides a solid starting point to build robust classifiers. However, training the classifiers to detect fine-grained labels of privacy policies’ segments requires a labeled dataset. For that purpose, we leverage the *Online Privacy Policies (OPP-115)* dataset, introduced by Wilson *et al.* [11]. This dataset contains 115 privacy policies manually annotated by skilled annotators (law school students). In total, the dataset has 23K annotated data practices. The annotations were at two levels. First, paragraph-sized segments were annotated according to one or more of the 10 high-level categories in Fig. 3 (e.g., *First Party Collection*, *Data Retention*). Then, annotators selected parts of the segment and annotated them using attribute–value pairs, e.g., *information_type: location*, *purpose: advertising*, etc. In total, there were 20 distinct attributes and 138 distinct values across all attributes. Of these, 122 values had more than 20 labels. In Fig. 3, we only show the mandatory attributes that should be present in all segments. Due to space limitation, we only show samples of the values for selected attributes in Fig. 3.

4.3 Hierarchical Multi-label Classification

To account for the multiple granularity levels in the policies’ text, we build a hierarchy of classifiers that are individually trained on handling specific parts of the problem.

At the **top level**, a classifier predicts one or more high-level categories of the input segment x (categories are the top-level, shaded boxes of Fig. 3). We train a multi-label classifier that provides us with the probability $p(c_i|x)$ of the occurrence of each high-level category c_i , taken from the set of all categories \mathcal{C} . In addition to allowing multiple categories per segment, using a multi-label classi-

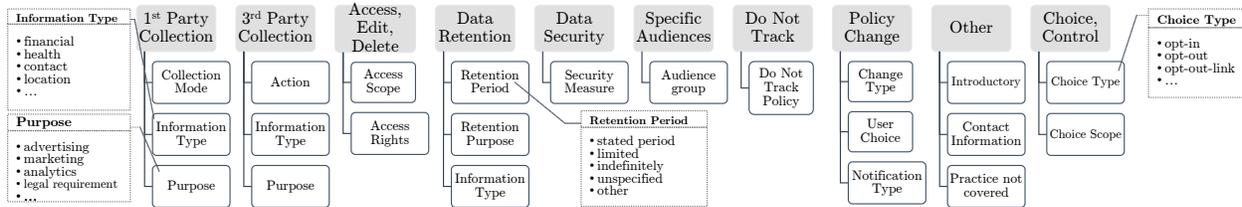


Fig. 3: The privacy taxonomy of Wilson *et al.* [11]. The top level of the hierarchy (shaded blocks) defines high-level privacy categories. The lower level defines a set of privacy attributes, each assuming a set of values. We show examples of values for some of the attributes.

fier makes it possible to determine whether a category is present in a segment by simply comparing its classification probability to a threshold of 0.5.

At the **lower level**, a set of classifiers predicts one or more values for each privacy attribute (the leaves in the taxonomy of Fig. 3). We train a set of multi-label classifiers on the attribute-level. Each classifier produces the probabilities $p(v_j|x)$ for the values $v_j \in \mathcal{V}(b)$ of a single attribute b . For example, given the attribute $b=information_type$, the corresponding classifier outputs the probabilities for elements in $\mathcal{V}(b)$: $\{financial, location, user\ profile, health, demographics, cookies, contact\ information, generic\ personal\ information, unspecified, \dots\}$.

An important consequence of this hierarchy is that interpreting the output of the attribute-level classifier depends on the categories’ probabilities. For example, the values’ probabilities of the attribute “*retention_period*” are irrelevant when the dominant high-level category is “*policy_change*.” Hence, for a category c_i , one would only consider the attributes descending from it in the hierarchy. We denote these attributes as $\mathcal{A}(c_i)$ and the set of all values across these attributes as $\mathcal{V}(c_i)$.

We use Convolutional Neural Networks (CNNs) internally within all the classifiers for two main reasons, which are also common in similar classification tasks. First, CNNs enable us to integrate pre-trained word embeddings that provide the classifiers with better generalization capabilities. Second, CNNs recognize when a certain set of tokens are a good indicator of the class, in a way that is invariant to their position within the input segment.

We use a similar CNN architecture for classifiers on both levels as shown in Fig. 4. Segments are split into tokens, using PENN Treebank tokenization in NLTK [29]. The embeddings layer outputs the word vectors of these tokens. We froze that layer, preventing its weights from being updated, in order to preserve the learnt semantic similarity between all the words present in our Policies Embeddings. Next, the word vectors pass through a Convolutional layer, whose main role is applying a non-linear function (a Rectified Linear Unit (ReLU)) over

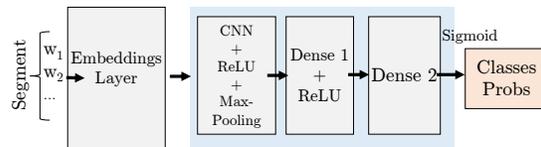


Fig. 4: Components of the CNN-based classifier used.

windows of k words. Then, a max-pooling layer combines the vectors resulting from the different windows into a single vector. This vector then passes through the first dense (i.e., fully-connected) layer with a ReLU activation function, and finally through the second dense layer. A *sigmoid* operation is applied to the output of the last layer to obtain the probabilities for the possible output classes. We used *multi-label cross-entropy loss* as the classifier’s objective function. We refer interested readers to [30] for further elaborations on how CNNs are used in such contexts.

Models’ Training. In total, we trained 20 classifiers at the attribute level (including the optional attributes). We also trained two classifiers at the category level: one for classifying segments and the other for classifying free-form queries. For the former, we include all the classes in Fig. 3. For the latter, we ignore the “*Other*” category as it is mainly for introductory sentences or uncovered practices [11], which are not applicable to users’ queries. For training the classifiers, we used the data from 65 policies in the OPP-115 dataset, and we kept 50 policies as a testing set. The hyper-parameters for each classifier were obtained by running a randomized grid-search. In Table 1, we present the evaluation metrics on the testing set for the category classifier intended for free-form queries. In addition to the precision, recall and F1 scores (macro-averaged per label³), we also show the top-1 precision metric, representing the fraction of segments where the top predicted category label oc-

³A successful multilabel classifier should not only predict the *presence* of a label, but also its *absence*. Otherwise, a model that predicts that all labels are present would have 100% precision and recall. For that, the precision in the table represents the macro-average of the precision in predicting the presence of each label and predicting its absence (similarly for recall and F1 metrics).

Table 1: Classification results for user queries at the category level. Hyperparameters: Embeddings size: 300, Number of filters: 200, Filter Size: 3, Dense Layer Size: 100, Batch Size: 40

Category	Prec.	Recall	F1	Top-1 Prec.	Support
1 st Party Collection	0.80	0.80	0.80	0.80	1267
3 rd Party Sharing	0.81	0.81	0.81	0.86	963
User Choice/Control	0.76	0.73	0.75	0.81	455
Data Security	0.87	0.86	0.87	0.77	202
Specific Audiences	0.95	0.94	0.95	0.91	156
Access, Edit, Delete	0.94	0.75	0.82	0.97	134
Policy Change	0.96	0.89	0.92	0.93	120
Data Retention	0.79	0.67	0.71	0.60	93
Do Not Track	0.97	0.97	0.97	0.94	16
Average	0.87	0.83	0.84	0.84	

curs in the annotators’ ground-truth labels. As evident in the table, our classifiers can predict the top-level privacy category with high accuracy. Although we consider the problem in the multi-label setting, these metrics are significantly higher than the models presented in the original OPP-115 paper [11]. The full results for the rest of classifiers are presented in the Appendix. The efficacy of these classifiers is further highlighted through queries that directly leverage their output in the applications described next.

5 Application Layer

Leveraging the power of the ML Layer’s classifiers, Polisis supports both *structured* and *free-from* queries about a privacy policy’s content. A structured query is a combination of first-order logic predicates over the predicted privacy classes and the policy segments, such as: $\exists s (s \in \text{policy} \wedge \text{information_type}(s)=\text{location} \wedge \text{purpose}(s) = \text{marketing} \wedge \text{user_choice}(s)=\text{opt-out})$. On the other hand, a free-form query is simply a natural language question posed directly by the users, such as “do you share my location with third parties?”. The response to a query is the set of segments satisfying the predicates in the case of a structured query or matching the user’s question in the case of a free-form query. The Application Layer builds on these query types to enable an array of applications for different privacy stakeholders. We take an exemplification approach to give the reader a better intuition on these applications, before delving deeper into two of them in the next sections.

Users: Polisis can automatically populate several of the previously-proposed short notices for privacy policies, such as nutrition tables and privacy icons [3, 18, 31, 32]. This task can be achieved by mapping the notices to a set of structured queries (*cf.* Sec. 6). Another possible application is privacy-centered comparative shopping [33]. A user can build on Polisis’ output to automatically quantify the privacy utility of a certain policy.

For example, such a privacy metric could be a combination of positive scores describing privacy-protecting features (e.g., policy containing a segment with the label: *retention_period: stated period*) and negative scores describing privacy-infringing features (e.g., policy containing a segment with the label: *retention_period: unlimited*). A major advantage of automatically generating short notices is that they can be seamlessly refreshed when policies are updated or when the rules to generate these notices are modified. Otherwise, discrepancies between policies and notices might arise over time, which deters companies from adopting the short notices in the first place.

By answering free-form queries with relevant policy segments, Polisis can remove the interface barrier between the policy and the users, especially in conversational interfaces (e.g., voice assistants and chatbots). Taking a step further, Polisis’ output can be potentially used to automatically rephrase the answer segments to a simpler language. A rule engine can generate text based on the combination of predicted classes of an answer segment (e.g., “We share data with third parties. This concerns our users’ information, like your online activities. We need this to respond to requests from legal authorities”).

Researchers: The difficulty of analyzing the data-collection claims by companies at scale has often been cited as a limitation in ecosystem studies (e.g., [34]). Polisis can provide the means to overcome that. For instance, researchers interested in analyzing apps that admit collecting health data [35, 36] could utilize Polisis to query a dataset of app policies. One example query can be formed by joining the label *information_type: health* with the category of *First Party Collection* or *Third Party Sharing*.

Regulators: Numerous studies from regulators and law and public policy researchers have manually analyzed the permissiveness of compliance checks [21, 37]. The number of assessed privacy policies in these studies is typically in the range of tens of policies. For instance, the Norwegian Consumer Council has investigated the level of ambiguity in defining personal information within only 20 privacy policies [37]. Polisis can scale such studies by processing a regulator’s queries on large datasets. For example, with Polisis, policies can be ranked according to an automated ambiguity metric by using the *information_type* attribute and differentiating between the label *generic_personal_information* and other labels specifying the type of data collected. Similarly, this applies to frameworks such as Privacy Shield [12] and the GDPR [15], where issues such as limiting the data usage purposes should be investigated.

Table 2: The list of Disconnect icons with their description, our interpretation, and Polisis’ queries.

Icon	Disconnect Description	Disconnect Color Assignment	Interpretation as Labels	Automated Color Assignment
	Discloses whether data it collects about you is used in ways other than you would reasonably expect given the site’s service?	Red: Yes, w/o choice to opt-out. Or, undisclosed. Yellow: Yes, with choice to opt-out. Green: No.	Let S be the segments with category: <i>first-party-collection-use</i> and purpose: <i>advertising</i> .	Yellow: All segments in S have category: <i>user-choice-control</i> and choice-type \in [<i>opt-in, opt-out-link, opt-out-via-contacting-company</i>] Green: $S = \phi$ Red: Otherwise
	Discloses whether it allows other companies like ad providers and analytics firms to track users on the site?	Red: Yes, w/o choice to opt-out. Or, undisclosed. Yellow: Yes, with choice to opt-out. Green: No.	Let S be the segments with category: <i>third-party-sharing-collection</i> , purpose: \in [<i>advertising, analytics-research</i>], and action-third-party \in [<i>track-on-first-party-website-app, collect-on-first-party-website-app</i>].	
	Discloses whether the site or service tracks a user’s actual geolocation?	Red: Yes, possibly w/o choice. Yellow: Yes, with choice. Green: No.	Let S be the segments with personal-information-type: <i>location</i> .	
	Discloses how long they retain your personal data?	Red: No data retention policy. Yellow: 12+ months. Green: 0-12 months.	Let S be the segments with category: <i>data-retention</i> .	Green: All segments in S have retention-period: \in [<i>stated-period, limited</i>]. Red: $S = \phi$ Yellow: Otherwise
	Has this website received TrustArc’s Children’s Privacy Certification?	Green: Yes. Gray: No.	Let S be the segments with category: <i>international-and-specific-audiences</i> and audience-type: <i>children</i>	Green: $\text{length}(S) > 0$ Red: Otherwise

Table 3: Prediction accuracy and κ for icon prediction, with the distribution of icons per color based on OPP-115 labels.

Icon	Accuracy	Cohen κ	Hellinger distance	N(R)	N(G)	N(Y)
Exp. Use	92%	0.76	0.12	41	8	1
Exp. Collection	88%	0.69	0.19	35	12	3
Precise Location	84%	0.68	0.21	32	14	4
Data Retention	80%	0.63	0.13	29	16	5
Children Privacy	98%	0.95	0.02	12	38	NA

6 Privacy Icons

Our first application shows the efficacy of Polisis in resolving structured queries to privacy policies. As a case study, we investigate the Disconnect privacy icons [18], described in the first three columns of Table 2. These icons evolved from a Mozilla-led working group that included the Electronic Frontier Foundation, Center for Democracy and Technology, and the W3C. The database powering these icons originated from TRUSTe (re-branded later as TrustArc), a privacy compliance company, which carried out the task of manually analyzing and labeling privacy policies.

In what follows, we first establish the accuracy of Polisis’ automatic assignment of privacy icons, using the Disconnect icons as a proof-of-concept. We perform a direct comparison between assigning these icons via Polisis and assigning them based on annotations by law students [11]. Second, we leverage Polisis to investi-

gate the level of permissiveness of the icons that Disconnect assigns based on the TRUSTe dataset. Our findings are consistent with the series of concerns raised around compliance-checking companies over the years [21, 38, 39]. This demonstrates the power of Polisis in scalable, automated auditing of privacy compliance checks.

6.1 Predicting Privacy Icons

Given that the rules behind the Disconnect icons are not precisely defined, we translated their description into explicit first-order logic queries to enable automatic processing. Table 2 shows the original description and color assignment provided by Disconnect. We also show our interpretation of each icon in terms of labels present in the OPP-115 dataset and the automated assignment of colors based on these labels. Our goal is not to reverse-engineer the logic behind the creation of these icons but to show that we can automatically assign such icons with high accuracy, given a plausible interpretation. Hence, this represents our best effort to reproduce the icons, but these rules could easily be adapted as needed.

To evaluate the efficacy of automatically selecting appropriate privacy icons, we compare the icons produced with Polisis’ automatic labels to the icons produced based on the law students’ annotations from the OPP-115 dataset [11]. We perform the evaluation over the same set of 50 privacy policies which we did not use to train Polisis (i.e., kept aside as a testing set). Each segment in the OPP-115 dataset has been labeled by three

experts. Hence, we take the union of the experts' labels on one hand and the predicted labels from Polisis on the other hand. Then, we run the logic presented in Table 2 (Columns 4 and 5) to assign icons to each policy based on each set of labels.

Table 3 shows the accuracy obtained per icon, measured as the fraction of policies where the icon based on *automatic labels* matched the icon based on the *experts' labels*. The average accuracy across icons is 88.4%, showing the efficacy of our approach in matching the experts' aggregated annotations. This result is significant in view of Miyazaki and Krishnamurthy's finding [21]: the level of agreement among 3 *trained human judges* assessing privacy policies ranged from 88.3% to 98.3%, with an average of 92.7% agreement overall. We also show Cohen's κ , an agreement measure that accounts for agreement due to random chance⁴. In our case, the values indicate *substantial to almost perfect* agreement [40]. Finally, we show the distribution of icons based on the *experts' labels* alongside Hellinger distance⁵, which measures the difference between that distribution and the one produced using the *automatic labels*. This distance assumes small values, illustrating that the distributions are very close. Overall, these results support the potential of automatically assigning privacy icons with Polisis.

6.2 Auditing Compliance Metrics

Given that we achieve a high accuracy in assigning privacy icons, it is intuitive to investigate how they compare to the icons assigned by Disconnect and TRUSTe. An important consideration in this regard is that several concerns have been raised earlier around the level of leniency of TRUSTe and other compliance companies [19, 20, 38, 39]. In 2000, the FTC conducted a study on privacy seals, including those of TRUSTe, and found that, of the 27 sites with a privacy seal, approximately only half implemented, at least in part, all four of the fair information practice principles and that only 63% implemented Notice and Choice. Hence, we pose the following question: *Can we automatically provide evidence of the level of leniency of the Disconnect icons using Polisis?* To answer this question, we designed an experiment to compare the icons extracted by Polisis' *automatic labels* to the icons assigned by Disconnect on real policies.

One obstacle we faced is that the Disconnect icons have been announced in June 2014 [41]; many privacy policies have likely been updated since then. To ensure that the privacy policies we consider are within a close time frame to those used by Disconnect, we make use of Ramanath *et al.*'s ACL/COLING 2014 dataset [42]. This

dataset contains the body of 1,010 privacy policies extracted between December 2013 and January 2014. We obtained the icons for the same set of sites using the Disconnect privacy icons extension [18]. Of these, 354 policies had been (at least partially) annotated in the Disconnect dataset. We automatically assign the icons for these sites by passing their policy contents into Polisis and applying the rules in Table 2 on the generated *automatic labels*. We report the results for the *Expected Use* and *Expected Collection* icons as they are directly interpretable by Polisis. We do not report the rest of the icons because the *location information* label in the OPP-115 taxonomy included non-precise location (e.g., zip codes), and there was no label that distinguishes the exact retention period. Moreover, the Children privacy icon is assigned through a certification process that does not solely rely on the privacy policy.

Fig. 5 shows the distribution of automatically extracted icons vs. the distribution of icons from Disconnect, when they were available. The discrepancy between the two distributions is obvious: the vast majority of the Disconnect icons have a yellow label, indicating that the policies offer the user an opt-out choice (from unexpected use or collection). The Hellinger distances between those distributions are 0.71 and 0.61 for *Expected Use* and *Expected Collection*, respectively (i.e., 3–5x the distance in the Table 3).

This discrepancy might stem from our icon-assignment strategy in Table 2, where we assign a yellow label only when “All segments in S (the concerned subset)” include the opt-in/opt-out choice, which could be considered as conservative. In Fig. 6, we show the icon distributions when relaxing the yellow-icon condition to become: “At least one segment in S ” includes the opt-in/opt-out choice. Intuitively, this means that the choice segment, when present, should explicitly mention advertising/analytics (depending on the icon type). Although the number of yellow icons increases slightly, the icons with the new permissive strategy are significantly red-dominated. The Hellinger distances between those distributions drop to 0.47 and 0.50 for *Expected Use* and *Expected Collection*, respectively. This result indicates that the majority of policies do not provide users a choice within the same segments describing data usage for advertising or data collection by third parties.

We go one step further to follow an even more permissive strategy where we assign the yellow label to any policy with $S! = \emptyset$, given that there is at least one segment in the whole policy (i.e., even outside S) with opt-in/opt-out choice. For example, a policy where third-party advertising is mentioned in the middle of the policy while the opt-out choice about another action is mentioned at the end of the policy would still receive a yellow label. The

⁴https://en.wikipedia.org/wiki/Cohen%27s_kappa

⁵https://en.wikipedia.org/wiki/Hellinger_distance

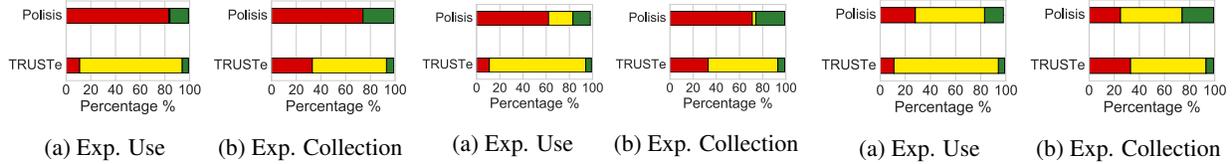


Fig. 5: Conservative icons’ interpretation Fig. 6: Permissive icons’ interpretation Fig. 7: Very permissive icons’ interpretation

icon distributions, in this case, are illustrated in Fig. 7, with Hellinger distance of 0.22 for Expected Use and 0.19 for Expected Collection. Only in this interpretation of the icons would the distributions of Disconnect and Polis come within reasonable proximity. In order to delve more into the factors behind this finding, we conducted a manual analysis of the policies. We found that, due to the way privacy policies are typically written, data collection and sharing are discussed in dedicated parts of the policy, without mentioning user choices. The choices (mostly opt-out) are discussed in a separate section when present, and they cover a small subset of the collected/shared data. In several cases, these choices are neither about the unexpected use (i.e., advertising) nor unexpected collection by third parties (i.e., advertising/analytics). Although our primary hypothesis is that this is due to TRUSTe’s database being generally permissive, it can be partially attributed to a potential discrepancy between our versions of analyzed policies and the versions used by TRUSTe (despite our efforts to reduce this discrepancy).

6.3 Discussion

There was no loss of generality when considering only two of the icons; they provided the needed evidence of TRUSTe/TrustArc potentially following a permissive strategy when assigning icons to policies. A developer could still utilize Polis to extract the rest of the icons by either augmenting the existing taxonomy or by performing additional natural language processing on the segments returned by Polis. In the vast majority of the cases, whenever the icon definition is to be changed (e.g., to reflect a modification in the regulations), this change can be supported at the rules level, without modifying Polis itself. This is because Polis already predicts a comprehensive set of labels, covering a wide variety of rules.

Furthermore, by automatically generating icons, we do not intend to push humans completely out of the loop, especially in situations where legal liability issues might arise. Polis can assist human annotators by providing initial answers to their queries and the supporting evidence. In other words, it accurately flags the segments of interest to an annotator’s query so that the annotator can make a final decision.

7 Free-form Question-Answering

Our second application of Polis is PriBot, a system that enables free-form queries (in the form of user questions) on privacy policies. PriBot is primarily motivated by the rise of conversation-first devices, such as voice-activated digital assistants (e.g., Amazon Alexa and Google Assistant) and smartwatches. For these devices, the existing techniques of linking to a privacy policy or reading it aloud are not usable. They might require the user to access privacy-related information and controls on a different device, which is not desirable in the long run [8].

To support these new forms of services and the emerging need for automated customer support in this domain [43], we present PriBot as an intuitive and user-friendly method to communicate privacy information. PriBot answers free-form user questions from a previously unseen privacy policy, in real time and with high accuracy. Next, we formalize the problem of free-form privacy QA and then describe how we leverage Polis to build PriBot.

7.1 Problem Formulation

The input to PriBot consists of a user question q about a privacy policy. PriBot passes q to the ML layer and the policy’s link to the Data Layer. The ML layer probabilistically annotates q and each policy’s segments with the privacy categories and attribute-value pairs of Fig. 3.

The segments in the privacy policy constitute the pool of candidate answers $\{a_1, a_2, \dots, a_M\}$. A subset \mathcal{G} of the answer pool is the ground-truth. We consider an answer a_k as *correct* if $a_k \in \mathcal{G}$ and as *incorrect* if $a_k \notin \mathcal{G}$. If \mathcal{G} is empty, then no answers exist in the privacy policy.

7.2 PriBot Ranking Algorithm

Ranking Score: In order to answer the user question, PriBot ranks each potential answer⁶ a by computing a proximity score $s(q, a)$ between a and the question q . This is within the *Class Comparison* module of the Application Layer. To compute $s(q, a)$, we proceed as follows. Given the output of the *Segment Classifier*, an answer is represented as a vector:

$$\alpha = \{p(c_i|a)^2 \times p(v_j|a) \mid c_i \in \mathcal{C}, v_j \in \mathcal{V}(c_i)\}$$

⁶For notational simplicity, we henceforth use a to indicate an answer instead of a_k .

for categories $c_i \in \mathcal{C}$ and values $v_j \in \mathcal{V}(c_i)$ descending from c_i . Similarly, given the output of the *Query Analyzer*, the question is represented as:

$$\beta = \{p(c_i|q)^2 \times p(v_j|q) \mid c_i \in \mathcal{C}, v_j \in \mathcal{V}(c_i)\}$$

The category probability in both α and β is squared to put more weight on the categories at the time of comparison. Next, we compute a certainty measure of the answer’s high-level categorization. This measure is derived from the entropy of the normalized probability distribution (p_n) of the predicted categories:

$$cer(a) = 1 - (-\sum(p_n(c_i|a) \times \ln(p_n(c_i|a))) / \ln(|\mathcal{C}|)) \quad (1)$$

Akin to a dot product between two vectors, we compute the score $s(q, a)$ as:

$$s(q, a) = \frac{\sum_i(\beta_i \times \min(\beta_i, \alpha_i))}{\sum_i \beta_i^2} \times cer(a) \quad (2)$$

As answers are typically longer than the question and involve a higher number of significant features, this score prioritizes the answers containing significant features that are also significant in the question. The *min* function and the denominator are used to normalize the score within the range $[0, 1]$.

To illustrate the strength of PriBot and its answer-ranking approach, we consider the following question (posed by a Twitter user):

“Under what circumstances will you release to 3rd parties?”

Then, we consider two examples of ranked segments by PriBot. The first segment has a ranking score of 0.63: “Personal information will not be used or disclosed for purposes other than those for which it was collected, except with the consent of the individual or as required by law. . .” The second has a ranking score of 0: “All personal information collected by the TTC will be protected by using appropriate safeguards against loss, theft and unauthorized access, disclosure, copying, use or modification.”

Although both example segments share terms such as “personal” and “information,” PriBot ranks them differently. It accounts for the fact that the question and the first segment share the same high-level category: *3rd Party Collection* while the second segment is categorized under *Data Security*.

Confidence Indicator: The ranking score is an internal metric that specifies how close each segment is to the question, but does not relay PriBot’s certainty in reporting a correct answer to a user. Intuitively, the confidence in an answer should be low when (1) the answer is semantically far from the question (i.e., $s(q, a)$ is low), (2) the question is interpreted ambiguously by Polisis, (i.e., classified into multiple high-level categories resulting in a high classification entropy), or (3) when the question

contains unknown words (e.g., in a non-English language or with too many spelling mistakes). Taking into consideration these criteria, we compute a confidence indicator as follows:

$$conf(q, a) = s(q, a) * \frac{(cer(q) + frac(q))}{2} \quad (3)$$

where the categorization certainty measure $cer(q)$ is computed similarly to $cer(a)$ in Eq. (1), and $s(q, a)$ is computed according to Eq. (2). The fraction of known words $frac(q)$ is based on the presence of the question’s words in the vocabulary of our *Policies Embeddings*’ corpus.

Potentially Conflicting Answers Another challenge is displaying potentially conflicting answers to users. One answer could describe a general sharing clause while another specifies an exception (e.g., one answer specifies “share” and another specifies “do not share”). To mitigate this issue, we used the same CNN classifier of Sec. 4 and exploited the fact that the OPP-115 dataset had optional labels of the form: “does” vs. “does not” to indicate the presence or absence of sharing/collection. Our classifier had a cross-validation F1 score of 95%. Hence, we can use this classifier to detect potential discrepancies between the top-ranked answers. The UI of PriBot can thus highlight the potentially conflicting answers to the user.

8 PriBot Evaluation

We assess the performance of PriBot with two metrics: the *predictive accuracy* (Sec. 8.3) of its QA-ranking model and the *user-perceived utility* (Sec. 8.4) of the provided answers. This is motivated by research on the evaluation of recommender systems, where the model with the best accuracy is not always rated to be the most helpful by users [44].

8.1 Twitter Dataset

In order to evaluate PriBot with realistic privacy questions, we created a new privacy QA dataset. It is worth noting that we utilize this dataset for the purpose of testing PriBot, not for training it. Our requirements for this dataset were that it (1) must include free-form questions about the privacy policies of different companies and (2) must have a ground-truth answer for each question from the associated policy.

To this end, we collected, from Twitter, privacy-related questions users had tweeted at companies. This approach avoids subject bias, which is likely to arise when eliciting privacy-related questions from individuals, who will not pose them out of genuine need. In our collection methodology, we aimed at a QA test set of size between 100 and 200 QA pairs, as is the convention in similar human-annotated QA evaluation domains, such

as the Text REtrieval Conference (TREC) and SemEval-2015 [45, 46, 47].

To avoid searching for questions via biased keywords, we started by searching for reply tweets that direct the users to a company’s privacy policy (e.g., using queries such as “*filter:replies our privacy policy*” and “*filter:replies our privacy statement*”). We then backtracked these reply tweets to the (parent) question tweets asked by customers to obtain a set of 4,743 pairs of tweets, containing privacy questions but also substantial noise due to the backtracking approach. Following the best practices of noise reduction in computational social science, we automatically filtered the tweets to keep those containing question marks, at least four words (excluding links, hashtags, mentions, numbers and stop words), and a link to the privacy policy, leaving 260 pairs of question–reply tweets. This is an example of a tweet pair which was removed by the automatic filtering:

Question: “@Nixxit your site is very suspicious.”

Answer: “@elitelinux Updated it with our privacy policy. Apologies, but we’re not fully up yet and running shoe string.”

Next, two of the authors independently validated each of the tweets to remove question tweets (a) that were not related to privacy policies, (b) to which the replies are not from the official company account, and (c) with inaccessible privacy policy links in their replies. The level of agreement (Cohen’s Kappa) among both annotators for the labels *valid* vs. *invalid* was almost perfect ($\kappa = 0.84$) [40]. The two annotators agreed on 231 of the question tweets (of the 260), tagging 182 as *valid* and 49 as *invalid*. This is an example of a tweet pair which was annotated as invalid:

Question: “What is your worth then? You can’t do it? Nuts.”

Answer: “@skychief26 3/3 You can view our privacy policy at <http://t.co/ksmalK1WwY>. Thanks.”

This is an example of a tweet pair annotated as valid:

Question: “@myen Are Evernote notes encrypted at rest?”

Answer: “We’re not encrypting at rest, but are encrypting in transit. Check out our Privacy Policy here: <http://bit.ly/1tauyfh>.”

As we wanted to evaluate the answers to these questions with a user study, our estimates of an adequately-sized study led us to randomly sample 120 tweets out of the tweets which both annotators labeled as valid questions. We henceforth refer to them as the *Twitter QA Dataset*.

8.2 QA Baselines

We compare PriBot’s QA model against three baseline approaches that we developed: (1) **Retrieval** reflects the state-of-the-art in term-matching retrieval algorithms, (2) **SemVec** representing a single neural network classifier,

and (3) **Random** as a control approach where questions are answered with random policy segments.

Our first baseline, Retrieval, builds on the BM25 algorithm [48], which is the state-of-the-art in ranking models employing term-matching. It has been used successfully across a range of search tasks, such as the TREC evaluations [49]. We improve on the basic BM25 model by computing the inverse document frequency on the *Policies Corpus* of Sec. 4.2 instead of a single policy. Retrieval ranks the segments in the policy according to their similarity score with the user’s question. This score depends on the presence of distinctive words that link a user’s question to an answer.

Our second baseline, SemVec employs a *single* classifier trained to distinguish among all the (mandatory) attribute-values (with > 20 annotations) from the OPP-115 dataset (81 classes in total). An example segment is “geographic location information or other location-based information about you and your device”. We obtain a micro-average precision of 0.56 (i.e., the classifier is, on average, predicting the right label across the 81 classes in 56% of the cases – compared to 3.6% precision for a random classifier). After training this model, we extract a “*semantic vector*”: a representation vector that accounts for the distribution of attribute values in the input text. We extract this vector as the input to the second dense layer (shown Fig. 4). SemVec ranks the similarity between a question and a policy segment using the Euclidean distance between semantic vectors. This approach is similar to what has been applied previously in image retrieval, where image representations learned from a large-scale image classification task were effective in visual search applications [50].

8.3 Predictive Accuracy Evaluation

Here, we evaluate the *predictive accuracy* of PriBot’s QA model by comparing its predicted answers against expert-generated ground-truth answers for the questions of the Twitter QA Dataset.

Ground-Truth Generation: Two of the authors generated the ground-truth answers to the questions from the Twitter QA Dataset. They were given a user’s question (tweet) and the segments of the corresponding policy. Each policy consists of 45 segments on average (*min=12, max=344, std=37*). Each annotator selected *independently*, the subset of these segments which they consider as best responding to the user’s question. This annotation took place *prior* to generating the answers using our models to avoid any bias. While deciding on the answers, the annotators accounted for the fact that multiple segments of the policy might answer a question.

After finishing the individual annotations, the two annotators consolidated the differences in their labels to reach an agreed-on set of segments; each assumed to be

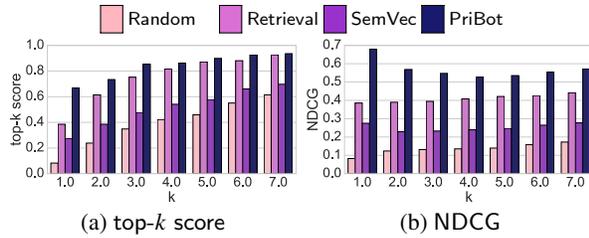


Fig. 8: Accuracy metrics as a function of k .

answering the question. We call this the *ground-truth* set for each question. The annotators agreed on at least one answer in 88% of the questions for which they found matching segments, thus signifying a substantial overlap. Cohen’s κ , measuring the agreement on one or more answer, was 0.65, indicating substantial agreement [40]. We release this dataset, comprising the questions, the policy segments, and the ground-truth answers per question at <https://pribot.org/data.html>.

We then generated, for each question, the predicted ranked list of answers according to each QA model (PriBot and the other three baselines). In what follows, we evaluate the predictive accuracy of these models.

Top- k Score: We first report the top- k score, a widely used and easily interpretable metric, which denotes the portion of questions having at least one correct answer in the top k returned answers. It is desirable to achieve a high top- k score for low values of k so that the user has to process less information before reaching a correct answer. Fig. 8a shows how the top- k score varies as a function of k . PriBot’s model has the best performance over the other three models by a large margin, especially at the low values of k . For example, at $k = 1$, PriBot has a top- k score of 0.68, which is significantly larger than the scores of 0.39 (Retrieval), 0.27 (SemVec), and 0.08 (Random) (p -value < 0.05 according to pairwise Fisher’s exact test, corrected with Bonferroni method for multiple comparisons). PriBot further reaches a top- k score of 0.75, 0.82, and 0.87 for $k \in \{2, 3, 4\}$. To put these numbers in the wider context of free-form QA systems, we note that the top-1 accuracy reported by IBM Watson’s team on a large insurance domain dataset (a training set of 12,889 questions and 21,325 answers) was 0.65 in 2015 [51] and was later improved to 0.69 in 2016 [52]. Given that PriBot had to overcome the absence of publicly available QA datasets, our top-1 accuracy value of 0.68 is on par with such systems. We also observe that the Retrieval model outperforms the SemVec model. This result is not entirely surprising since we seeded Retrieval with a large corpus of 130K unsupervised policies, thus improving its performance on answers with matching terms.

Policy Length We now assess the impact of the policy length on PriBot’s accuracy. First, we report the *Nor-*

malized Discounted Cumulative Gain (NDCG) [53]. Intuitively, it indicates that a relevant document’s usefulness decreases logarithmically with the rank. This metric captures how presenting the users with more choices affects their user experience as they need to process more text. Also, it is not biased by the length of the policy. The *DCG* part of the metric is computed as $DCG_k = \sum_{i=1}^k \frac{rel_i}{\log_2(i+1)}$, where rel_i is 1 if answer a_i is correct and 0 otherwise. NDCG at k is obtained by normalizing the DCG_k with the maximum possible DCG_k across all values of k . We show in Fig. 8b the average NDCG across questions for each value of k . It is clear that PriBot’s model consistently exhibits superior NDCG. This indicates that PriBot is poised to perform better in a system where low values of k matter the most.

Second, to further focus on the effect of policy length, we categorize the policy lengths (*#segments*) into *short*, *medium*, and *high*, based on the 33rd and the 66th percentiles (i.e., corresponding to *#segments* of 28 and 46). We then compute a metric independent of k , namely, the Mean Average Precision (MAP), which is the mean of the area under the precision-recall curve across all questions. Informally, MAP is an indicator of whether all the correct answers get ranked highly. We see from Fig. 9 that, for short policies, the Retrieval model is within 15% of the MAP of PriBot’s model, which makes sense given the smaller number of potential answers. With medium-sized policies, PriBot’s model is better by a large margin. This margin is still considerable with long policies.

Confidence Indicator Comparing the confidence (using the indicator from Eq. (3)) of incorrect answers predicted by PriBot (mean=0.37, variance=0.04) with the confidence of correct answers (mean=0.49, variance=0.05) shows that PriBot places lower confidence in the answers that turn out to be incorrect. Hence, we can use the confidence indicator to filter out the incorrect answers. For example, by setting the condition: $conf(q, a) \geq 0.6$ to accept PriBot’s answers, we can enhance the top-1 accuracy to 70%. This indicator delivers another advantage: its components are independently interpretable by the application logic. If the score $s(q, a)$ of the top-1 answer is too low, the user can be notified that the policy might not contain an answer to the question. A low value of $cer(q)$ indicates that the user might have asked an ambiguous question; the system can ask the user back for a clarification.

Pre-trained Embeddings Choice As discussed in Sec. 4, we utilize our custom Policies Embeddings, which have the two properties of (1) being domain-specific and (2) using subword embeddings to handle out-of-vocabulary words. We test the efficacy of this choice by studying three variants of pre-trained embeddings. For the first variant, we start from our Policies Embeddings (PE), and we disable the subwords mode,

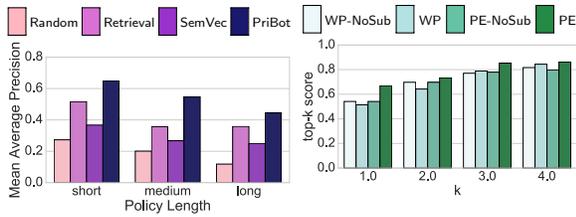


Fig. 9: Variation of MAP across policy lengths.

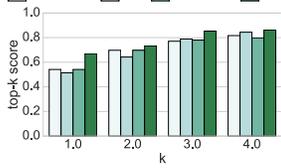
thus only satisfying the first property; we call it PE-NoSub. The second variant is the *fastText* Wikipedia Embeddings from [54], trained on the English Wikipedia, thus only satisfying the second property; we denote it as WP. The third variant is WP, with the subword mode disabled, thus satisfying neither property; we call it WP-NoSub. In Fig. 10, we show the top- k score of PriBot on our Twitter QA dataset with each of the four pre-trained embeddings. First, we can see that our Policies Embeddings outperform the other models for all values of k , scoring 14% and 5% more than the closest variant at $k = 1$ and $k = 2$, respectively. As expected, the domain-specific model without subwords embeddings (PE-NoSub) has a weaker performance by a significant margin, especially for the top-1 answer. Interestingly, the difference is much narrower between the two Wikipedia embeddings since their vocabulary already covers more than 2.5M tokens. Hence, subword embeddings play a less pronounced role there. In sum, the advantage of using subwords embeddings with the PE model originates from their domain specificity and their ability to compensate for the missing words from the vocabulary.

8.4 User-Perceived Utility Evaluation

We conducted a user study to assess the *user-perceived utility* of the automatically generated answers. This assessment was done for each of the four different conditions (Retrieval, SemVec, PriBot and Random). We evaluated the top-3 responses of each QA approach to each question. Thus, we assess the utility of 360 answers to 120 questions per approach.

Study Design: We used a between-subject design by constructing four surveys, each corresponding to a different evaluation condition. We display a series of 17 QA pairs (each on a different page). Of these, 15 are a random subset of the pool of 360 QA pairs (of the evaluated condition) such that a participant does not receive two QA pairs with the same question. The other two questions are randomly positioned anchor questions serving as attention checkers. Additionally, we enforce a minimum duration of 15 seconds for the respondent to evaluate each QA pair, with no maximum duration enforced. We include an open-ended Cloze reading comprehension

Fig. 10: top- k score of PriBot with different pre-trained embeddings.



Question: Hey @HostGator can you not sell my phone number to telemarketers? Paying for privacy protection should protect me from YOU TOO

Candidate Response: Public Forums. Please remember that any information you may disclose or post on public areas of our websites or the Internet, becomes public information. You should exercise caution when deciding to disclose personal information in these public areas.

Does the candidate response provide an answer to the given question?

- Definitely Yes:** It perfectly answers the question.
- Partially Yes:** It answers the bulk of the question, though there might be more to say.
- Undecided:** I find it too difficult to give a judgment on this pair.
- Partially No:** It doesn't answer the question; only has a slight clue.
- Definitely No:** It totally misses the topic of the question.

Fig. 11: An example of a QA pair displayed to the respondents.

test [55]; we used the test to weed out the responses with a low score, indicating a poor reading skill.

Participant Recruitment: After obtaining an IRB approval, we recruited 700 Amazon MTurk workers with previous success rate $>95\%$, to complete our survey. With this number of users, each QA pair received evaluations from at least 7 different individuals. We compensated each respondent with \$2. With an average completion time of 14 minutes, this makes the average pay around \$8.6 per hour (US Federal minimum wage is \$7.25). While not fully representative of the general population, our set of participants exhibited high intra-group diversity, but little difference across the respondent groups. Across all respondents, the average age is 34 years ($std=10.5$), 62% are males, 38% are females, more than 82% are from North America, more than 87% have some level of college education, and more than 88% reported being employed.

QA Pair Evaluation: To evaluate the relevance for a QA pair, we display the question and the candidate answer as shown in Fig. 11. We asked the respondents to rate whether the candidate response provides an answer to the question on a 5-point Likert scale (1=*Definitely Yes* to 5=*Definitely No*), as evident in Fig. 11. We denote a respondent's evaluation of a **single** candidate answer corresponding to a QA pair as relevant (irrelevant) if s/he chooses either *Definitely Yes* (*Definitely No*) or *Partially Yes* (*Partially No*). We consolidate the evaluations of multiple users per answer by following the methodology outlined in similar studies [10], which consider the answer as relevant if labeled as relevant by a certain fraction of users. We took this fraction as 50% to ensure a majority agreement. Generally, we observed the respondents to agree on the relevance of the answers. Highly mixed responses, where 45–55% of the workers tagged the answer as relevant, constituted less than 16% of the cases.

User Study Results: As in the previous section, we compute the top- k score for relevance (i.e., the portion of questions having at least one user-relevant answer in the top k returned answers). Table 4 shows this score for

Table 4: top- k relevance score by evaluation group.

Group	N	top- k Relevance Score		
		$k = 1$	$k = 2$	$k = 3$
Random	180	0.37	0.59	0.76
Retrieval	184	0.46	0.71	0.79
SemVec	153	0.48	0.71	0.85
PriBot	183	0.70	0.78	0.89

the four QA approaches with $k \in \{1, 2, 3\}$, where PriBot clearly outperforms the three baseline approaches. The respondents regarded at least one of the top-3 answers as relevant for 89% of the questions, with the first answer being relevant in 70% of the cases. In comparison, for $k = 1$, the scores were 46% and 48% for the Retrieval and the SemVec models respectively (p -value ≤ 0.05 according to pairwise Fishers exact test, corrected with Holm-Bonferroni method for multiple comparisons). An avid reader might notice some differences between the predictive models' accuracy (Section 8.3) and the users' perceived quality. This is actually consistent with the observations from research in recommender systems where the prediction accuracy does not always match user's satisfaction [44]. For example, the top- k score metric for accuracy differs by 2%, -3%, and 6% with respect to the perceived relevance in the PriBot model. Another example is that the SemVec model and the Retrieval have smaller differences in this study than Sec. 8.3. We conjecture that the score shift with SemVec model is due to some users accepting answers which match the question's topic even when the actual details of the answer are irrelevant.

9 Discussion

Limitations Polisis might be limited by the employed privacy taxonomy. Although the OPP-115 taxonomy covers a wide variety of privacy practices [11], there are certain types of applications that it does not fully capture. One mitigation is to use Polisis as an initial step in order to filter the relevant data at a high level before applying additional, application-specific text processing. Another mitigation is to leverage Polisis' modularity by amending it with new categories/attributes and training these new classes on the relevant annotated dataset.

Moreover, Polisis, like any automated approach, exhibits instances of misclassification that should be accounted for in any application building on it. One way to mitigate this problem is using confidence scores, similar to that of Eq. (3) to convey the (un)certainty of a reported result, whether it is an answer, an icon, or another form of short notice. Last but not least, Polisis is not guaranteed to be robust in handling an adversarially constructed privacy policy. An adversary could include valid and meaningful statements in the privacy policy, carefully crafted

to mislead Polisis' automated classifiers. For example, an adversary can replace words, in the policy, with synonyms that are far in our embeddings space. While the modified policy has the same meaning, Polisis might misclassify the modified segments.

Deployment: We provide three prototype web applications for end-users. The first is an application that visualizes the different aspects in the privacy policy, powered by the annotations from Polisis (available as a web application and a browser extension for Chrome and Firefox). The second is a chatbot implementation of PriBot for answering questions about privacy policies in a conversational interface. The third is an application for extracting the privacy labels from several policies, given their links. These applications are available at <https://pribot.org>.

Legal Aspects We also want to stress the fact that Polisis *is not intended* to replace the legally-binding privacy policy. Rather, it offers a complementary interface for privacy stakeholders to easily inquire the contents of a privacy policy. Following the trend of automation in legal advice [56], insurance claim resolution [57], and privacy policy presentation [58, 16], third parties, such as automated legal services firms or regulators, can deploy Polisis as a solution for their users. As is the standard in such situations, these parties should amend Polisis with a disclaimer specifying that it is based on automatic analysis and does not represent the actual service provider [59].

Companies and service providers can internally deploy an application similar to PriBot as an assistance tool for their customer support agents to handle privacy-related inquiries. Putting the human in the loop allows for a favorable trade-off between the utility of Polisis and its legal implications. For a wider discussion on the issues surrounding automated legal analysis, we refer the interested reader to the works of McGinnis and Pearce [60] and Pasquale [61].

Privacy-Specificity of the Approach: Finally, our approach is uniquely tailored to the privacy domain both from the data perspective and from the model-hierarchy perspective. However, we envision that applications with similar needs would benefit from extensions of our approach, both on the classification level and the QA level.

10 Related Work

Privacy Policy Analysis: There have been numerous attempts to create easy-to-navigate and alternative presentations of privacy policies. Kelley *et al.* [32] studied using nutrition labels as a paradigm for displaying privacy notices. Icons representing the privacy policies have also been proposed [31, 62]. Others have proposed standards to push service providers to encode privacy policies in a machine-readable format, such as P3P [13], but they

have not been adopted by browser developers and service providers. Polisis has the potential to automate the generation of a lot of these notices, without relying on the respective parties to do it themselves.

Recently, several researchers have explored the potential of automated analysis of privacy policies. For example, Liu *et al.* [58] have used deep learning to model the vagueness of words in privacy policies. Zimmeck *et al.* [63] have been able to show significant inconsistencies between app practices and their privacy policies via automated analysis. These studies, among others [64, 65], have been largely enabled by the release of the OPP-115 dataset by Wilson *et al.* [11], containing 115 privacy policies extensively annotated by law students. Our work is the first to provide a generic system for the automated analysis of privacy policies. In terms of the comprehensiveness and the accuracy of the approach, Polisis makes a major improvement over the state of the art. It allows transitioning from labeling of policies with a few practices (e.g., the works by Zimmeck and Bellovin [16] and Sathyendra *et al.* [17]) to a much more fine-grained annotation (up to 10 high-level and 122 fine-grained classes), thus enabling a richer set of applications.

Evaluating the Compliance Industry: Regulators and researchers are continuously scrutinizing the practices of the privacy compliance industry [21, 38, 39]. Miyazaki and Krishnamurthy [21] found no support that participating in a seal program is an indicator of following privacy practice standards. The FTC has found discrepancies between the practical behaviors of the companies, as reported in their privacy policies, and the privacy seals they have been granted [39]. Polisis can be used by these researchers and regulators to automatically, and continuously perform such checks at scale. It can provide the initial evidence that could be processed by skilled experts afterward, thus reducing the analysis time and the cost.

Automated Question Answering: Our QA system, PriBot, is focused on *non-factoid* questions, which are usually complex and open-ended. Over the past few years, deep learning has yielded superior results to traditional retrieval techniques in this domain [51, 52, 66]. Our main contribution is that we build a QA system, without a dataset that includes questions and answers, while achieving results on par with the state of the art on other domains. We envision that our approach could be transplanted to other problems that face similar issues.

11 Conclusion

We proposed Polisis, the first generic framework that enables detailed automatic analysis of privacy policies. It can assist users, researchers, and regulators in processing and understanding the content of privacy policies at scale. To build Polisis, we developed a new hierarchy

of neural networks that extracts both high-level privacy practices as well as fine-grained information from privacy policies. Using this extracted information, Polisis enables several applications. In this paper, we demonstrated two applications: structured and free-form querying. In the first example, we use Polisis' output to extract short notices from the privacy policy in the form of privacy icons and to audit TRUSTe's policy analysis approach. In the second example, we build PriBot that answers users' free-form questions in real time and with high accuracy. Our evaluation of both applications reveals that Polisis matches the accuracy of expert analysis of privacy policies. Besides these applications, Polisis opens opportunities for further innovative privacy policy presentation mechanisms, including summarizing policies into simpler language. It can also enable comparative shopping applications that advise the consumer by comparing the privacy aspects of multiple applications they want to choose from.

Acknowledgements

This research was partially funded by the Wisconsin Alumni Research Foundation and the US National Science Foundation under grant agreements CNS-1330596 and CNS-1646130.

References

- [1] F. H. Cate, "The limits of notice and choice," *IEEE Security Privacy*, vol. 8, no. 2, pp. 59–62, March 2010.
- [2] Federal Trade Commission, "Protecting Consumer Privacy in an Era of Rapid Change," March 2012.
- [3] J. Gluck, F. Schaub, A. Friedman, H. Habib, N. Sadeh, L. F. Cranor, and Y. Agarwal, "How short is too short? implications of length and framing on the effectiveness of privacy notices," in *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*. Denver, CO: USENIX Association, 2016, pp. 321–340.
- [4] A. M. McDonald and L. F. Cranor, "The cost of reading privacy policies," *ISJLP*, vol. 4, p. 543, 2008.
- [5] President's Council of Advisors on Science and Technology, "Big data and privacy: A technological perspective. Report to the President, Executive Office of the President," May 2014.
- [6] F. Schaub, R. Balebako, and L. F. Cranor, "Designing effective privacy notices and controls," *IEEE Internet Computing*, vol. 21, no. 3, pp. 70–77, 2017.
- [7] Federal Trade Commission, "Internet of Things, Privacy & Security in a Connected World," Jan. 2015.
- [8] F. Schaub, R. Balebako, A. L. Durity, and L. F. Cranor, "A design space for effective privacy notices," in *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*. Ottawa: USENIX Association, 2015, pp. 1–17.
- [9] A. Rao, F. Schaub, N. Sadeh, A. Acquisti, and R. Kang, "Expecting the unexpected: Understanding mismatched privacy expectations online," in *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*. Denver, CO: USENIX Association, 2016, pp. 77–96.

- [10] S. Wilson, F. Schaub, R. Ramanath, N. Sadeh, F. Liu, N. A. Smith, and F. Liu, "Crowdsourcing annotations for websites' privacy policies: Can it really work?" in *Proceedings of the 25th International Conference on World Wide Web*, ser. WWW '16. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2016, pp. 133–143.
- [11] S. Wilson, F. Schaub, A. A. Dara, F. Liu, S. Cherivirala, P. G. Leon, M. S. Andersen, S. Zimmeck, K. M. Sathyendra, N. C. Russell, T. B. Norton, E. H. Hovy, J. R. Reidenberg, and N. M. Sadeh, "The creation and analysis of a website privacy policy corpus," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.
- [12] U.S. Department of Commerce, "Privacy shield program overview," <https://www.privacyshield.gov/Program-Overview>, 2017, accessed: 10-01-2017.
- [13] L. Cranor, *Web privacy with P3P*. "O'Reilly Media, Inc.", 2002.
- [14] P. G. Kelley, J. Bresee, L. F. Cranor, and R. W. Reeder, "A "nutrition label" for privacy," in *Proceedings of the 5th Symposium on Usable Privacy and Security*, ser. SOUPS '09. New York, NY, USA: ACM, 2009, pp. 4:1–4:12.
- [15] "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)," *Official Journal of the European Union*, vol. L119, pp. 1–88, May 2016.
- [16] S. Zimmeck and S. M. Bellovin, "Privee: An architecture for automatically analyzing web privacy policies." in *USENIX Security*, vol. 14, 2014.
- [17] K. M. Sathyendra, S. Wilson, F. Schaub, S. Zimmeck, and N. Sadeh, "Identifying the provision of choices in privacy policy text," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2764–2769.
- [18] Disconnect, "Privacy Icons," <https://web.archive.org/web/20170709022651/disconnect.me/icons>, accessed: 07-01-2017.
- [19] B. Edelman, "Adverse selection in online "trust" certifications," in *Proceedings of the 11th International Conference on Electronic Commerce*, ser. ICEC '09. New York, NY, USA: ACM, 2009, pp. 205–212.
- [20] T. Foremski, "TRUSTe responds to Facebook privacy problems..." <http://www.zdnet.com/article/truste-responds-to-facebook-privacy-problems/>, 2017, accessed: 2017-10-01.
- [21] A. D. Miyazaki and S. Krishnamurthy, "Internet seals of approval: Effects on online privacy policies and consumer perceptions," *Journal of Consumer Affairs*, vol. 36, no. 1, pp. 28–49, 2002.
- [22] G. Glavaš, F. Nanni, and S. P. Ponzetto, "Unsupervised text segmentation using semantic relatedness graphs," in **SEM 2016: The Fifth Joint Conference on Lexical and Computational Semantics: proceedings of the conference; August 11-12 2016, Berlin, Germany*. Stroudsburg, Pa.: Association for Computational Linguistics, 2016, pp. 125–130.
- [23] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2014, pp. 1746–1751.
- [24] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.
- [25] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.
- [26] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification." in *ACL (1)*, 2014, pp. 1555–1565.
- [27] N. Viennot, E. Garcia, and J. Nieh, "A measurement study of google play," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, no. 1. ACM, 2014, pp. 221–233.
- [28] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.
- [29] S. Bird and E. Loper, "Nltk: the natural language toolkit," in *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. Association for Computational Linguistics, 2004, p. 31.
- [30] D. Britz, "Understanding convolutional neural networks for NLP," <http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/>, 2015, accessed: 01-01-2017.
- [31] L. F. Cranor, P. Guduru, and M. Arjula, "User interfaces for privacy agents," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 13, no. 2, pp. 135–178, 2006.
- [32] P. G. Kelley, J. Bresee, L. F. Cranor, and R. W. Reeder, "A nutrition label for privacy," in *Proceedings of the 5th Symposium on Usable Privacy and Security*. ACM, 2009, p. 4.
- [33] J. Y. Tsai, S. Egelman, L. Cranor, and A. Acquisti, "The effect of online privacy information on purchasing behavior: An experimental study," *Information Systems Research*, vol. 22, no. 2, pp. 254–268, 2011.
- [34] A. Razaghpahan, R. Nithyanand, N. Vallina-Rodriguez, S. Sundaresan, M. Allman, and C. K. P. Gill, "Apps, trackers, privacy, and regulators," in *25th Annual Network and Distributed System Security Symposium, NDSS 2018*, 2018.
- [35] A. Aktypi, J. Nurse, and M. Goldsmith, "Unwinding ariadne's identity thread: Privacy risks with fitness trackers and online social networks," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017.
- [36] E. Steel and A. Dembosky, "Health apps run into privacy snags," *Financial Times*, 2013.
- [37] Norwegian Consumer Council, "Appfail report threats to consumers in mobile apps," Norwegian Consumer Council, Tech. Rep., 2016.
- [38] E. M. Caudill and P. E. Murphy, "Consumer online privacy: Legal and ethical issues," *Journal of Public Policy & Marketing*, vol. 19, no. 1, pp. 7–19, 2000.
- [39] R. Pitofsky, S. Anthony, M. Thompson, O. Swindle, and T. Leary, "Privacy online: Fair information practices in the electronic marketplace," *Statement of the Federal Trade Commission before the Committee on Commerce, Science and Transportation, United States Senate, Washington, DC*, 2000.
- [40] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *biometrics*, pp. 159–174, 1977.
- [41] TRUSTe, "TRUSTe & Disconnect Introduce Visual Icons to Help Consumers Understand Privacy Policies," <http://www.trustarc.com/blog/2014/06/23/truste-disconnect-introduce-visual-icons-to-help-consumers-understand-privacy-policies/>, June 2013, accessed: 07-01-2017.
- [42] R. Ramanath, F. Liu, N. M. Sadeh, and N. A. Smith, "Unsupervised alignment of privacy policies using hidden markov models," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, 2014, pp. 605–610.

- [43] C. Schneider, “10 reasons why ai-powered, automated customer service is the future,” <https://www.ibm.com/blogs/watson/2017/10/10-reasons-ai-powered-automated-customer-service-future>, October 2017, accessed: 10-01-2017.
- [44] B. P. Knijnenburg, M. C. Willemsen, and S. Hirtbach, “Receiving recommendations and providing feedback: The user-experience of a recommender system,” in *International Conference on Electronic Commerce and Web Technologies*. Springer, 2010, pp. 207–216.
- [45] H. T. Dang, D. Kelly, and J. J. Lin, “Overview of the trec 2007 question answering track,” in *Trec*, vol. 7, 2007, p. 63.
- [46] H. Llorens, N. Chambers, N. Mostafazadeh, J. Allen, and J. Pustejovsky, “Qa tempeval: Evaluating temporal information understanding with qa,”
- [47] M. Wang, N. A. Smith, and T. Mitamura, “What is the jeopardy model? a quasi-synchronous grammar for qa,” in *EMNLP-CoNLL*, vol. 7, 2007, pp. 22–32.
- [48] S. Robertson, “Understanding inverse document frequency: on theoretical arguments for IDF,” *Journal of Documentation*, vol. 60, pp. 503–520, 2004.
- [49] M. Beaulieu, M. Gatford, X. Huang, S. Robertson, S. Walker, and P. Williams, “Okapi at trec-5,” *NIST SPECIAL PUBLICATION SP*, pp. 143–166, 1997.
- [50] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki, “Visual instance retrieval with deep convolutional networks,” *arXiv preprint arXiv:1412.6574*, 2014.
- [51] M. Feng, B. Xiang, M. R. Glass, L. Wang, and B. Zhou, “Applying deep learning to answer selection: A study and an open task,” in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2015, Scottsdale, AZ, USA, December 13-17, 2015*, 2015, pp. 813–820.
- [52] M. Tan, C. dos Santos, B. Xiang, and B. Zhou, “Improved representation learning for question answer matching,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.
- [53] K. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of ir techniques,” *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.
- [54] Facebook, “Wiki word vectors,” <https://fasttext.cc/docs/en/pretrained-vectors.html>, 2017, accessed: 2017-10-01.
- [55] Cambridge English Language Assessment, *Cambridge English Proficiency Certificate of Proficiency in English CEFR level C2, Handbook for Teachers*. University of Cambridge, 2013.
- [56] J. Goodman, “Legal technology: the rise of the chatbots,” <https://www.lawgazette.co.uk/features/legal-technology-the-rise-of-the-chatbots/5060310.article>, 2017, accessed: 2017-04-27.
- [57] A. Levy, “Microsoft ceo satya nadella: for the future of chat bots, look at the insurance industry,” <http://www.cnbc.com/2017/01/09/microsoft-ceo-satya-nadella-bots-in-insurance-industry.html>, 2017, accessed: 2017-04-27.
- [58] F. Liu, N. L. Fella, and K. Liao, “Modeling language vagueness in privacy policies using deep neural networks,” in *2016 AAAI Fall Symposium Series*, 2016.
- [59] T. Hwang, “The laws of (legal) robotics,” Robot, Robot & Hwang LLP, Tech. Rep., 2013.
- [60] J. O. McGinnis and R. G. Pearce, “The great disruption: How machine intelligence. will transform the role of lawyers in the delivery of legal services,” *Fordham L. Rev.*, vol. 82, pp. 3041–3481, 2014.
- [61] F. Pasquale and G. Cashwell, “Four futures of legal automation,” *UCLA L. Rev. Discourse*, vol. 63, p. 26, 2015.
- [62] L.-E. Holtz, H. Zwingelberg, and M. Hansen, “Privacy policy icons,” in *Privacy and Identity Management for Life*. Springer, 2011, pp. 279–285.
- [63] S. Zimmeck, Z. Wang, L. Zou, R. Iyengar, B. Liu, F. Schaub, S. Wilson, N. Sadeh, S. M. Bellovin, and J. Reidenberg, “Automated analysis of privacy requirements for mobile apps,” in *24th Annual Network and Distributed System Security Symposium, NDSS 2017*, 2017.
- [64] F. Liu, S. Wilson, F. Schaub, and N. Sadeh, “Analyzing vocabulary intersections of expert annotations and topic models for data practices in privacy policies,” in *2016 AAAI Fall Symposium Series*, 2016.
- [65] K. M. Sathyendra, F. Schaub, S. Wilson, and N. Sadeh, “Automatic extraction of opt-out choices from privacy policies,” in *2016 AAAI Fall Symposium Series*, 2016.
- [66] J. Rao, H. He, and J. Lin, “Noise-contrastive estimation for answer selection with deep neural networks,” in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, ser. CIKM ’16. New York, NY, USA: ACM, 2016, pp. 1913–1916.

Appendix: Full Classification Results

We present the classification results at the category level for the Segment Classifier and at 15 selected attribute levels, using the hyperparameters of Table 1.

Classification results at the category level for the Segment Classifier

Label	Prec.	Recall	F1	Top-1 Prec.	Support
Data Retention	0.83	0.66	0.71	0.68	88
Data Security	0.88	0.83	0.85	0.79	201
Do Not Track	0.94	0.97	0.95	0.88	16
1 st Party Collection	0.79	0.79	0.79	0.79	1211
Specific Audiences	0.96	0.94	0.95	0.93	156
Introductory/Generic	0.81	0.66	0.70	0.75	369
Policy Change	0.95	0.84	0.88	0.93	112
Non-covered Practice	0.76	0.67	0.70	0.60	280
Privacy Contact Info	0.90	0.85	0.87	0.88	137
3 rd Party Sharing	0.79	0.80	0.79	0.82	908
Access, Edit, Delete	0.89	0.75	0.80	0.87	133
User Choice/Control	0.74	0.74	0.74	0.69	433
Average	0.85	0.79	0.81	0.80	

Classification results for attribute: *change-type*

Label	Prec.	Recall	F1	Support
privacy-relevant-change	0.78	0.76	0.77	77
unspecified	0.79	0.76	0.76	90
Average	0.78	0.76	0.76	

Classification results for attribute: *notification-type*

Label	Prec.	Recall	F1	Support
general-notice-in-privacy-policy	0.80	0.77	0.78	76
general-notice-on-website	0.64	0.62	0.62	52
personal-notice	0.69	0.66	0.67	50
unspecified	0.81	0.72	0.75	24
Average	0.73	0.69	0.71	

Classification results for attribute: *identifiability*

Label	Prec.	Recall	F1	Support
aggregated-or-anonymized	0.89	0.89	0.89	284
identifiable	0.81	0.81	0.81	492
unspecified	0.63	0.63	0.63	98
Average	0.77	0.78	0.77	

Classification results for attribute: *do-not-track-policy*

Label	Prec.	Recall	F1	Support
honored	1.00	1.00	1.00	8
not-honored	1.00	1.00	1.00	26
Average	1.00	1.00	1.00	

Classification results for attribute: *security-measure*

Label	Prec.	Recall	F1	Support
data-access-limitation	0.89	0.78	0.81	35
generic	0.84	0.83	0.83	102
privacy-review-audit	0.97	0.58	0.62	13
privacy-security-program	0.87	0.69	0.73	31
secure-data-storage	0.82	0.64	0.69	17
secure-data-transfer	0.91	0.80	0.84	26
secure-user-authentication	0.97	0.58	0.63	12
Average	0.90	0.70	0.74	

Classification results for attribute: *personal-information-type*

Label	Prec.	Recall	F1	Support
computer-information	0.84	0.80	0.82	88
contact	0.90	0.89	0.90	342
cookies-and-tracking-elements	0.95	0.92	0.94	272
demographic	0.93	0.90	0.92	86
financial	0.89	0.86	0.87	99
generic-personal-information	0.82	0.79	0.80	441
health	1.00	0.56	0.61	8
ip-address-and-device-ids	0.93	0.93	0.93	104
location	0.88	0.88	0.88	107
personal-identifier	0.67	0.61	0.63	31
social-media-data	0.73	0.84	0.78	23
survey-data	0.77	0.86	0.81	22
unspecified	0.71	0.70	0.71	456
user-online-activities	0.80	0.82	0.81	224
user-profile	0.79	0.68	0.72	96
Average	0.84	0.80	0.81	

Classification results for attribute: *purpose*

Label	Prec.	Recall	F1	Support
additional-service-feature	0.75	0.76	0.75	374
advertising	0.92	0.91	0.92	286
analytics-research	0.88	0.86	0.87	239
basic-service-feature	0.76	0.73	0.74	401
legal-requirement	0.92	0.91	0.91	79
marketing	0.86	0.83	0.84	312
merger-acquisition	0.95	0.96	0.95	38
personalization-customization	0.79	0.80	0.80	149
service-operation-and-security	0.81	0.77	0.79	200
unspecified	0.72	0.68	0.70	249
Average	0.84	0.82	0.83	

Classification results for attribute: *choice-type*

Label	Prec.	Recall	F1	Support
browser-device-privacy-controls	0.89	0.95	0.92	171
dont-use-service-feature	0.69	0.65	0.67	213
first-party-privacy-controls	0.75	0.62	0.66	71
opt-in	0.78	0.81	0.79	406
opt-out-link	0.82	0.74	0.77	167
opt-out-via-contacting-company	0.87	0.81	0.84	127
third-party-privacy-controls	0.82	0.62	0.66	99
unspecified	0.65	0.54	0.56	117
Average	0.78	0.72	0.73	

Classification results for attribute: *third-party-entity*

Label	Prec.	Recall	F1	Support
collect-on-first-party-website-app	0.78	0.64	0.68	113
receive-shared-with	0.87	0.87	0.87	843
see	0.83	0.79	0.81	63
track-on-first-party-website-app	0.75	0.86	0.79	107
unspecified	0.60	0.51	0.52	57
Average	0.77	0.74	0.73	

Classification results for attribute: *access-type*

Label	Prec.	Recall	F1	Support
edit-information	0.65	0.62	0.63	172
unspecified	0.98	0.64	0.71	14
view	0.55	0.53	0.53	47
Average	0.73	0.60	0.62	

Classification results for attribute: *audience-type*

Label	Prec.	Recall	F1	Support
californians	0.98	0.97	0.98	60
children	0.98	0.97	0.97	161
europeans	0.97	0.95	0.96	23
Average	0.98	0.97	0.97	

Classification results for attribute: *choice-scope*

Label	Prec.	Recall	F1	Support
both	0.61	0.53	0.54	71
collection	0.74	0.68	0.70	302
first-party-collection	0.63	0.55	0.56	109
first-party-use	0.80	0.68	0.71	236
third-party-sharing-collection	0.81	0.60	0.64	98
third-party-use	0.57	0.51	0.50	60
unspecified	0.55	0.55	0.55	76
use	0.62	0.55	0.56	140
Average	0.67	0.58	0.59	

Classification results for attribute: *action-first-party*

Label	Prec.	Recall	F1	Support
collect-in-mobile-app	0.84	0.75	0.79	68
collect-on-mobile-website	0.58	0.54	0.56	11
collect-on-website	0.65	0.65	0.65	739
unspecified	0.61	0.60	0.60	294
Average	0.67	0.64	0.65	

Classification results for attribute: *does-does-not*

Label	Prec.	Recall	F1	Support
does	0.82	0.93	0.86	1436
does-not	0.82	0.93	0.86	200
Average	0.82	0.93	0.86	

Classification results for attribute: *retention-period*

Label	Prec.	Recall	F1	Support
indefinitely	0.45	0.48	0.47	8
limited	0.74	0.75	0.75	27
stated-period	0.94	0.94	0.94	10
unspecified	0.82	0.77	0.77	41
Average	0.74	0.74	0.73	