Privee: An Architecture for Automatically Analyzing Web Privacy Policies

Sebastian Zimmeck and Steven M. Bellovin, *Columbia University*

https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/zimmeck

This paper is included in the Proceedings of the
23rd USENIX Security Symposium.

August 20–22, 2014 • San Diego, CA

ISBN 978-1-931971-15-7

Privee: An Architecture for Automatically Analyzing Web Privacy Policies

Sebastian Zimmeck and Steven M. Bellovin

Department of Computer Science, Columbia University
{*sebastian,smb*}@*cs.columbia.edu*

Abstract

Privacy policies on websites are based on the notice-and-choice principle. They notify Web users of their privacy choices. However, many users do not read privacy policies or have difficulties understanding them. In order to increase privacy transparency we propose Privee—a software architecture for analyzing essential policy terms based on crowdsourcing and automatic classification techniques. We implement Privee in a proof of concept browser extension that retrieves policy analysis results from an online privacy policy repository or, if no such results are available, performs automatic classifications. While our classifiers achieve an overall F-1 score of 90%, our experimental results suggest that classifier performance is inherently limited as it correlates to the same variable to which human interpretations correlate—the ambiguity of natural language. This finding might be interpreted to call the notice-and-choice principle into question altogether. However, as our results further suggest that policy ambiguity decreases over time, we believe that the principle is workable. Consequently, we see Privee as a promising avenue for facilitating the notice-and-choice principle by accurately notifying Web users of privacy practices and increasing privacy transparency on the Web.

1 Introduction

Information privacy law in the U.S. and many other countries is based on the free market notice-and-choice principle [28]. Instead of statutory laws and regulations, the privacy regime is of a contractual nature—the provider of a Web service posts a privacy policy, which a user accepts by using the site. In this sense, privacy policies are fundamental building blocks of Web privacy. The Federal Trade Commission (FTC) strictly enforces companies' violations of their promises in privacy policies. However, only few users read privacy policies and those who do find them oftentimes hard to understand [58]. The resulting information asymmetry leaves users uninformed about their privacy choices [58], can lead to market failure [57], and ultimately casts doubt on the notice-and-choice principle.

Various solutions were proposed to address the problem. However, none of them gained widespread acceptance—neither in the industry, nor among users. Most prominently, The Platform for Privacy Preferences (P3P) project [29, 32] was not widely adopted, mainly, because of a lack of incentive on part of the industry to express their policies in P3P format. In addition, P3P was also criticized for not having enough expressive power to describe privacy practices accurately and completely [28, 11]. Further, existing crowdsourcing solutions, such as Terms of Service; Didn't Read (ToS;DR) [5], may not scale well and still need to gain more popularity. Informed by these experiences, which we address in more detail in Section 2, we present Privee—a novel software architecture for analyzing Web privacy policies. In particular, our contributions are:

- the Privee concept that combines rule and machine learning (ML) classification with privacy policy crowdsourcing for seamless integration into the existing privacy regime on the Web (Section 3);
- an implementation of Privee in a Google Chrome browser extension that interacts with privacy policy websites and the ToS;DR repository of crowdsourced privacy policy results (Section 4);
- a statistical analysis of our experimental results showing that the ambiguity of privacy policies makes them inherently difficult to understand for both humans and automatic classifiers (Section 5);
- pointers for further research on notice-and-choice and adaptations that extend Privee as the landscape of privacy policy analysis changes and develops (Section 6).

2 Related Work

While only few previous works are directly applicable, our study is informed by four areas of previous research: privacy policy languages (Section 2.1), legal information extraction (Section 2.2), privacy policy crowdsourcing (Section 2.3), and usable privacy (Section 2.4).

2.1 Privacy Policy Languages

Initial work on automatic privacy policy analysis focused on making privacy policies machine-readable. That way a browser or other user agent could read the policies and alert the user of good and bad privacy practices. Reidenberg [67] suggested early on that Web services should represent their policies in the Platform for Internet Content Selection (PICS) format [10]. This and similar suggestions lead to the development of P3P [29, 32], which provided a machine-readable language for specifying privacy policies and displaying their content to users [33]. To that end, the designers of P3P implemented various end users tools, such as Privacy Bird [30], a browser extension for Microsoft's Internet Explorer that notifies users of the privacy practices of a Web service whose site they visit, and Privacy Bird Search [24], a P3P-enabled search engine that returns privacy policy information alongside search results.

The development of P3P was complemented by various other languages and tools. Of particular relevance was A P3P Preference Privacy Exchange Language (APPEL) [31], which enabled users to express their privacy preferences vis-à-vis Web services. APPEL was further extended in the XPath project [14] and inspired the User Privacy Policy (UPP) language [15] for use in social networks. For industry use, the Platform for Enterprise Privacy Practices (E-P3P) [47] was developed allowing service providers to formulate, supervise, and enforce privacy policies. Similar languages and frameworks are the Enterprise Privacy Authorization Language (EPAL) [18], the SPARCLE Policy Workbench [22, 23], Jeeves [78], and XACML [12]. However, despite all efforts the adoption rate of P3P policies among Web services remained low [11], and the P3P working group was closed in 2006 due to lack of industry participation [28].

Instead of creating new machine-readable privacy policy formats we believe that it is more effective to use what is already there—privacy policies in natural language. The reasons are threefold: First, natural language is the de-facto standard for privacy policies on the Web, and the P3P experience shows that there is currently no industry-incentive to move to a different standard. Second, U.S. governmental agencies are in strong support of the natural language format. In particular, the FTC, the main privacy regulator, called for more industry-efforts to increase policy standardization and comprehensibility [38]. Another agency, the National Science Foundation, awarded $3.75 million to the Usable Privacy Policy Project [9] to explore possibilities of automatic policy analysis. Third, natural language has stronger expressive power compared to a privacy policy language. It allows for industry-specific formulation of privacy practices and accounts for the changing legal landscape over time.

2.2 Legal Information Extraction

Given our decision to make use of natural language policies, the question becomes how salient information can be extracted from unordered policy texts. While most works in legal information extraction relate to domains other than privacy, they still provide some guidance. For example, Westerhout et al. [75, 76] had success in combining a rule-based classifier with an ML classifier to identify legal definitions. In another line of work de Maat et al. [35, 36] aimed at distinguishing statutory provisions according to types (such as procedural rules or appendices) and patterns (such as definitions, rights, or penal provisions). They concluded that it was unnecessary to employ something more complex than a simple pattern recognizer [35, 36]. Other tasks focused on the extraction of information from statutory and regulatory laws [21, 20], the detection of legal arguments [59], or the identification of case law sections [54, 71].

To our knowledge, the only works in the privacy policy domain are those by Ammar et al. [16], Costante et al. [26, 27], and Stamey and Rossi [70]. As part of the Usable Privacy Policy Project [9] Ammar et al. presented a pilot study [16] with a narrow focus on classifying provisions for the disclosure of information to law enforcement officials and users' rights to terminate their accounts. They concluded the feasibility of natural language analysis in the privacy policy domain in general. In their first work [26] Costante et al. used general natural language processing libraries to evaluate the suitability of rule-based identification of different types of user information that Web services collect. Their results are promising and indicate the feasibility of rule-based classifiers. In a second work [27] Costante et al. selected an ML approach for assessing whether privacy policies cover certain subject matters. Finally, Stamey and Rossi [70] provided a program for identifying ambiguous words in privacy policies.

The discussed works [16, 26, 27, 70] confirm the suitability of rule and ML classifiers in the privacy policy domain. However, neither provides a comprehensive concept, nor addresses, for example, how to process the policies or how to make use of crowdsourcing results. The latter point is especially important because, as shown in Section 5, automatic policy classification on its own is in-

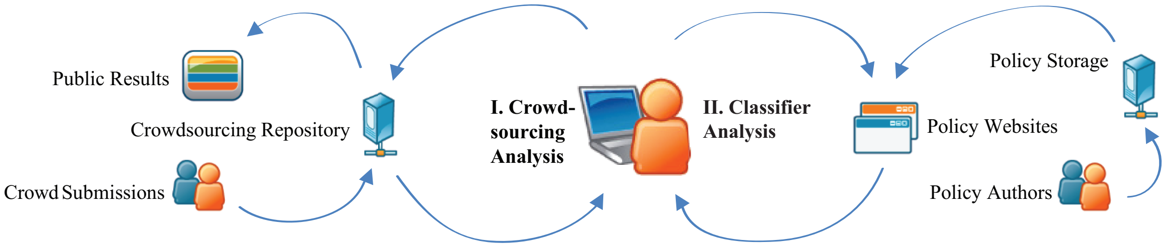


Figure 1: Privee overview. When a user requests a privacy policy analysis, the program checks whether the analysis results are available at a crowdsourcing repository (to which crowd contributors can submit analysis results of policies). If results are available, they are returned and displayed to the user (I. Crowdsourcing Analysis). If no results are available, the policy text is fetched from the policy website, analyzed by automatic classifiers on the client machine, and then the analysis results are displayed to the user (II. Classifier Analysis).

herently limited. In addition, as the previous works' purpose is to generally show the viability of natural language privacy policy analysis, they are constrained to classifying one or two individual policy terms or features. As they process each classification task separately, there was also no need to address questions of handling multiple classifiers or discriminating which extracted features belong to which classification task. Because of their limited scope none of the previous works relieves the user from actually reading the analyzed policy. In contrast, it is our goal to provide users with a privacy policy summary in lieu of the full policy. We want to condense a policy into essential terms, make it more comprehensible, provide guidance on the analyzed practices, and give an overall evaluation of its privacy level.

2.3 Privacy Policy Crowdsourcing

There are various crowdsourcing repositories where crowd contributors evaluate the content of privacy policies and submit their results into a centralized collection for publication on the Web. Sometimes policies are also graded. Among those repositories are ToS;DR [5], privacychoice [4], TOSBack [7], and TOSBack2 [8]. Crowdsourcing has the advantage that it combines the knowledge of a large number of contributors, which, in principle, can lead to much more nuanced interpretations of ambiguous policy provisions than current automatic classifiers could provide. However, all crowdsourcing approaches suffer from a lack of participation and, consequently, do not scale well. While the analysis results of the most popular websites may be available, those for many lesser known sites are not. In addition, some repositories only provide the possibility to look up the results on the Web without offering convenient user access, for example, by means of a browser extension or other software.

2.4 Usable Privacy

Whether the analysis of a privacy policy is based on crowdsourcing or automatic classifications, in order to notify users of the applicable privacy practices it is not enough to analyze policy content, but rather the results must also be presented in a comprehensible, preferably, standardized format [60]. In this sense, usable privacy is orthogonal to the other related areas: no matter how the policies are analyzed, a concise, user-friendly notification is always desirable. In particular, privacy labels may help to succinctly display privacy practices [48, 49, 51, 65, 66]. Also, privacy icons, such as those proposed by PrimeLife [39, 45], KnowPrivacy [11], and the Privacy Icons project [3], can provide visual clues to users. However, care must be taken that the meaning of the icons is clear to the users [45]. In any case, it should be noted that while usability is an important element of the Privee concept, we have not done a usability study for our Privee extension as it is just a proof of concept.

3 The Privee Concept

Figure 1 shows a conceptual overview of Privee. Privee makes use of automatic classifiers and complements them with privacy policy crowdsourcing. It integrates all components of the current Web privacy ecosystem. Policy authors write their policies in natural language and do not need to adopt any special machine-readable policy format. While authors certainly can express the same semantics as with P3P, which we demonstrate in Section 4.6.2, they can also go beyond and use their language much more freely and naturally.

When a user wants to analyze a privacy policy, Privee leverages the discriminative power of crowdsourcing. As we will see in Section 5 that classifiers and human interpretations are inherently limited by ambiguous language, it is especially important to resolve those ambiguities by providing a forum for discussion and developing consensus among many crowd contributors. Further, Privee complements the crowdsourcing analysis with the ubiquitous applicability of rule and ML classifiers for policies that are not yet analyzed by the crowd. Because the computational requirements are low, as shown in Section 5.3, a real time analysis is possible.

As the P3P experience showed [28] that a large fraction of Web services with P3P policies misrepresented their privacy practices, presumably in order to prevent user agents from blocking their cookies, any privacy policy analysis software must be guarded against manipulation. However, natural language approaches, such as Privee, have an advantage over P3P and other machine-readable languages. Because it is not clear whether P3P policies are legally binding [69] and the FTC never took action to enforce them [55], the misrepresentation of privacy practices in those policies is a minor risk that many Web services are willing to take. This is true for other machine-readable policy solutions as well. In contrast, natural language policies can be valid contracts [1] and are subject to the FTC's enforcement actions against unfair or deceptive acts or practices (15 U.S.C. §45(a)(1)). Thus, we believe that Web services are more likely to ensure that their natural language policies represent their practices accurately.

Given that natural language policies attempt to truly reflect privacy practices, it is important that the policy text is captured completely and without additional text, in particular, free from advertisements on the policy website. Further, while it is true that an ill-intentioned privacy policy author might try to deliberately use ambiguous language to trick the classifier analysis, this strategy can only go so far as ambiguous contract terms are interpreted against the author (Restatement (Second) of Contracts, §206) and might also cause the FTC to challenge them as unfair or deceptive. Beyond safeguarding the classifier analysis, it is also important to prevent the manipulation of the crowdsourcing analysis. In this regard, the literature on identifying fake reviews should be brought to bear. For example, Wu et al. [77] showed that fake reviews can be identified by a suspicious grade distribution and their posting time following negative reviews. In order to ensure that the crowdsourcing analysis returns the latest results the crowdsourcing repository should also keep track of privacy policy updates.

4 The Privee Browser Extension

We implemented Privee as a proof of concept browser extension for Google Chrome (version 35.0.1916.153). Figure 2 shows a simplified overview of the program flow. We wrote our Privee extension in JavaScript using

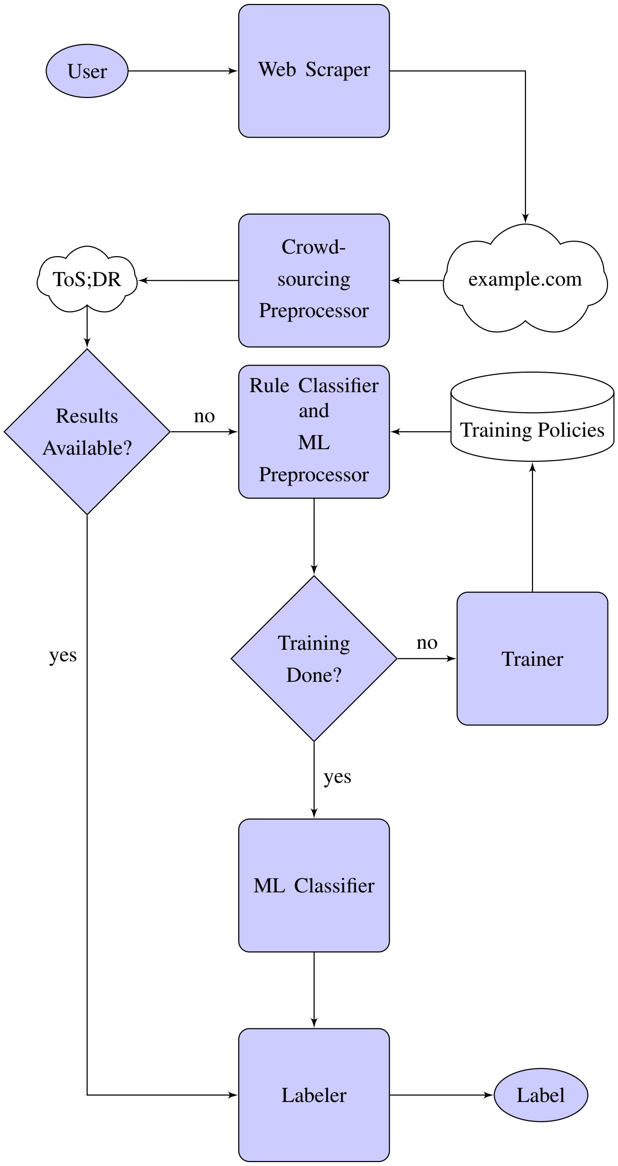


Figure 2: Simplified program flow. After the user has started the extension, the Web scraper obtains the text of the privacy policy to be analyzed (example.com) as well as the current URL (http://example.com/). The crowdsourcing preprocessor then extracts from the URL the ToS;DR identifier and checks the ToS;DR repository for results. If results are available, they are retrieved and forwarded to the labeler, which converts them to a label for display to the user. However, if no results are available on ToS;DR the policy text is analyzed. First, the rule classifier attempts a rule-based classification. However, if that is not possible the ML preprocessor prepares the ML classification. It checks if the ML classifier is already trained. If that is the case, the policy is classified by the ML classifier, assigned a label according to the classifications, and the results are displayed to the user. Otherwise, a set of training policies is analyzed by the trainer first and the program proceeds to the ML classifier and labeler afterwards. The set of training policies is included in the extension package and only needs to be analyzed for the first run of the ML classifier. Thereafter, the training results are kept in persistent storage until deletion by the user.

the jQuery library and Ajax functions for client-server communication. While we designed our extension as an end user tool, it can also be used for scientific or industrial research, for example, in order to easily compare different privacy policies to each other. In this Section we describe the various stages of program execution.

4.1 Web Scraper

The user starts the Privee extension by clicking on its icon in the Chrome toolbar. Then, the Web scraper obtains the text of the privacy policy that the user wants to analyze and retrieves the URL of the user's current website. While the rule and ML classifier analysis only works from the site that contains the policy to be analyzed, the crowdsourcing analysis works on any website whose URL contains the policy's ToS;DR identifier.

4.2 Crowdsourcing Preprocessor

The crowdsourcing preprocessor is responsible for managing the interaction with the ToS;DR repository. It receives the current URL from the Web scraper from which it extracts the ToS;Dr identifier. It then connects to the API of ToS;DR and checks for the availability of analysis results, that is, short descriptions of privacy practices and sometimes an overall letter grade. The results, if any, are forwarded to the labeler and displayed to the user. Then the extension terminates. Otherwise, the policy text, which the crowdsourcing preprocessor also received from the Web scraper, is forwarded to the rule classifier and ML preprocessor.

4.3 Rule Classifier and ML Preprocessor

Generally, classifiers can be based on rule or ML algorithms. In our preliminary experiments we found that for some classification categories a rule classifier worked better, in others an ML classifier, and in others again a combination of both [71, 76]. We will discuss our classifier selection in Section 5.1 in more detail. In this Section we will focus on the feature selection process for our rule classifier and ML preprocessor. Both rule classification and ML preprocessing are based on feature selection by means of regular expressions.

Our preliminary experiments revealed that classification performance depends strongly on feature selection. Ammar et al. [16] discuss a similar finding. Comparable to other domains [76], feature selection is particularly useful in our case for avoiding misclassifications due to the heavily imbalanced structure of privacy policies. For example, in many multi-page privacy policies there is often only one phrase that determines whether the Web service is allowed to combine the collected information with information from third parties to create personal profiles of users. Especially, supervised ML classifiers do not work well in such cases, even with undersampling (removal of uninteresting examples) or oversampling (duplication of interesting examples) [52]. Possible solutions to the problem are the separation of policies into different content zones and applying a classifier only to relevant content zones [54] or—the approach we adopted—running a classifier only on carefully selected features.

Our extension's feature selection process begins with the removal of all characters from the policy text that are not letters or whitespace and conversion of all remaining characters to lower case. However, the positions of removed punctuations are preserved because, as noted by Biagoli et al. [19], a correct analysis of the meaning of legal documents often depends on the position of punctuation. In order to identify the features that are most characteristic for a certain class we used the term frequency-inverse document frequency (tf-idf) statistic as a proxy. The tf-idf statistic measures how concentrated into relatively few documents the occurrences of a given word are in a document corpus [64]. Thus, words with high tf-idf values correlate strongly with the documents in which they appear and can be used to identify topics in that document that are not discussed in other documents. However, instead of using individual words as features we observed that the use of bigrams lead to better classification performance, which was also discussed in previous works [16, 59].

```
(ad|advertis.*) (compan.*|network.*|provider.*|
servin.*|serve.*|vendor.*)|(behav.*|context.*|
network.*|parti.*|serv.*) (ad|advertis.*)
```

Listing 1: Simplified pseudocode of the regular expression to identify whether a policy allows advertising tracking. For example, the regular expression would match "contextual advertising."

The method by which our Privee extension selects characteristic bigrams, which usually consist of two words, but can also consist of a word and a punctuation mark, is based on regular expressions. It applies a three-step process that encompasses both rule classification and ML preprocessing. To give an example, for the question whether the policy allows advertising tracking (e.g., by ad cookies) the first step consists of trying to match the regular expression in Listing 1, which identifies bigrams that nearly always indicate that advertising tracking is allowed. If any bigram in the policy matches, no further analysis happens, and the policy is classified by the rule classifier as allowing advertising tracking. If the regular expression does not match, the second step attempts to extract further features that can be associated with advertising tracking (which are, however, more general than the previous ones). Listing 2 shows the regular expression used for the second step.

```
(ad|advertis|market) (.+)|(.+) (ad|advertis|
market)
```

Listing 2: Simplified pseudocode of the regular expression to extract relevant phrases for advertising tracking. For example, the regular expression would match "no advertising."

The second step—the ML preprocessing—is of particular importance for our analysis because it prepares classification of the most difficult cases. It extracts the features on which the ML classifier will run later. To that end, it first uses the Porter stemmer [63] to reduce words to their morphological root [19]. Such stemming has the effect that words with common semantics are clustered together [41]. For example, "collection," "collected," and "collect" are all stemmed into "collect." As a side note, while stemming had some impact, we did not find a substantial performance increase for running the ML classifier on stemmed features compared to unstemmed features. In the third step, if no features were extracted in the two previous steps, the policy is classified as not allowing advertising tracking.

4.4 Trainer

In the training stage our Privee extension checks whether the ML classifier is already trained. If that is not the case, a corpus of training policies is preprocessed and analyzed. The analysis of a training policy is similar to the analysis of a user-selected policy, except that the extension does not check for crowdsourcing results and only applies the second and third step of the rule classifier and ML preprocessor phase. The trainer's purpose is to gather statistical information about the features in the training corpus in order to prepare the classification of the user-selected policy. It stores the training results locally in the user's browser memory using persistent Web storage, which is, in principle, similar to cookie storage.

4.5 Training Data

The training policies are held in a database that is included in the extension package. The database holds a total of 100 training policies. In order to obtain a representative cross section of training policies, we selected the majority of our policies randomly from the Alexa top 500 websites for the U.S. [6] across various domains (banking, car rental, social networking, etc.). However, we also included a few random policies from lesser frequented U.S. sites and sites from other countries that published privacy policies in English. The trainer accesses these training policies one after another and adds the training results successively to the client's Web storage. After all results are added the ML classifier is ready for classification.

4.6 ML Classifier

We now describe the ML classifier design (Section 4.6.1) and the classification categories (Section 4.6.2).

4.6.1 ML Classifier Design

In order to test the suitability of different ML algorithms for analyzing privacy policies we performed preliminary experiments using the Weka library [43]. Performance for the different algorithms varied. We tested all algorithms available on Weka, among others the Sequential Minimal Optimization (SMO) algorithm with different kernels (linear, polynomial, radial basis function), random forest, J48 (C4.5), IBk nearest neighbor, and various Bayesian algorithms (Bernoulli naive Bayes, multinomial naive Bayes, Bayes Net). Surprisingly, the Bayesian algorithms were among the best performers. Therefore, we implemented naive Bayes in its Bernoulli and multinomial version. Because the multinomial version ultimately proved to have better performance, we settled on this algorithm.

As Manning et al. [56] observed, naive Bayes classifiers have good accuracy for many tasks and are very efficient, especially, for high-dimensional vectors, and they have the advantage that training and classification can be accomplished with one pass over the data. Our naive Bayes implementation is based on their specification [56]. In general, naive Bayes classifiers make use of Bayes' theorem. The probability, P, of a document, d, being in a category, c, is

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c), \tag{1}$$

where $P(c)$ is the prior probability of a document occurring in category c, n_d is the number of terms in d that are used for the classification decision, and $P(t_k|c)$ is the conditional probability of term t_k occurring in a document of category c [56]. In other words, $P(t_k|c)$ is interpreted as a measure of how much evidence t_k contributes for c being the correct category [56]. The best category to select for a document in a naive Bayes classification is the category for which it holds that

$$\arg\max_{c \in \mathbb{C}} \hat{P}(c|d) = \arg\max_{c \in \mathbb{C}} \hat{P}(c) \prod_{1 \leq k \leq n_d} \hat{P}(t_k|c), \tag{2}$$

where $\mathbb{C}$ is a set of categories, which, in our case, is always of size two (e.g., {ad tracking, no ad tracking}).

The naive assumption is that the probabilities of individual terms within a document are independent of each other given the category [41]. However, our implementation differs from the standard implementation and tries to alleviate the independence assumption. Instead of processing individual words of the policies we try to capture some context by processing bigrams.

Analyzing the content of a privacy policy requires multiple classification decisions. For example, the classifier has to decide whether personal information can be collected, disclosed to advertisers, retained indefinitely, and so on. This type of classification is known as multi-label classification because each analyzed document can receive more than one label. One commonly used approach for multi-label classification with L labels consists of dividing the task into $|L|$ binary classification tasks [74]. However, other solutions handle multi-label data directly by extending specific learning algorithms [74]. We found it simpler to implement the first approach. Specifically, at execution time we create multiple classifier instances—one for each classification category—by running the classifier on category-specific features extracted by the ML preprocessor.

4.6.2 Classification Categories

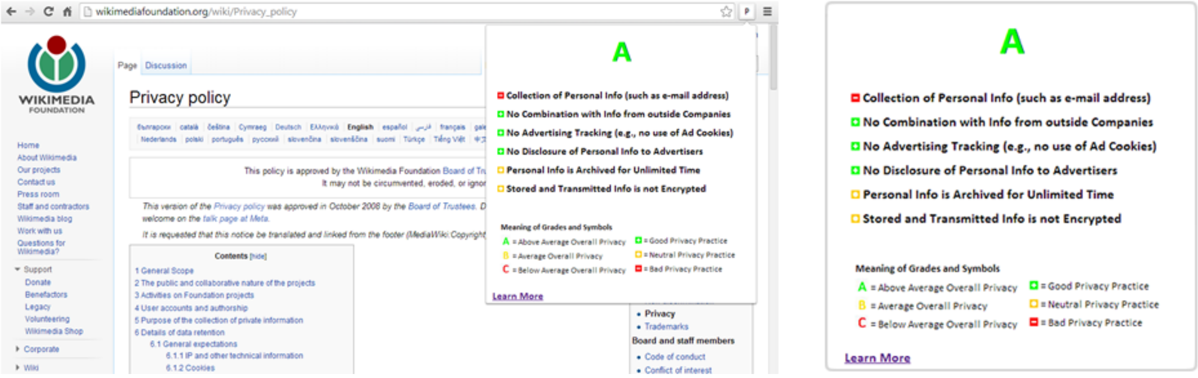
For which types of information should privacy policies actually be analyzed? In answering this question, one starting point are fair information practices [25]. Another one are the policies themselves. After all, while it is true that privacy law in the U.S. generally does not require policies to have a particular content, it can be observed that all policies conventionally touch upon four different themes: information collection, disclosure, use, and management (management refers to the handling of information, for example, whether information is encrypted). The four themes can be analyzed on different levels of abstraction. For example, for disclosure of information, it could simply be analyzed whether information is disclosed to outside parties in general, or it could be investigated more specifically whether information is disclosed to service providers, advertisers, governmental agencies, credit bureaus, and so on.

At this point it should be noted that not all information needs to be analyzed. In some instances privacy policies simply repeat mandatory law without creating any new rights or obligations. For example, a federal statute in the U.S.—18 U.S.C. §2703(c)(1)(A) and (B)—provides that the government can demand the disclosure of customer information from a Web service provider after obtaining a warrant or suitable court order. As this law applies independently of a privacy policy containing an explicit statement to that end, the provision that the provider will disclose information to a governmental entity under the requirements of the law can be inferred from the law itself. In fact, even if a privacy policy states to the contrary, it should be assumed that such information disclosure will occur. Furthermore, if privacy policies stay silent on certain subject matters, default rules might apply and fill the gaps.

Another good indicator of what information should be classified is provided by user studies. According to one study [30], knowing about sharing, use, and purpose of information collection is very important to 79%, 75%, and 74% of users, respectively. Similarly, in another study [11] users showed concern for the types of personal information collected, how personal information is collected, behavioral profiling, and the purposes for which the information may be used. While it was only an issue of minor interest earlier [30], the question how long a company keeps personal information about its users is a topic of increasing importance [11]. Based on these findings, we decided to perform six different binary classifications, that is, whether or not a policy

- allows collection of personal information from users (Collection);
- provides encryption for information storage or transmission (Encryption);
- allows ad tracking by means of ad cookies or other trackers (Ad Tracking);
- restricts archiving of personal information to a limited time period (Limited Retention);
- allows the aggregation of information collected from users with information from third parties (Profiling);
- allows disclosure of personal information to advertisers (Ad Disclosure).

For purposes of our analysis, where applicable, it is assumed that the user has an account with the Web service whose policy is analyzed and is participating in any offered sweepstakes or the like. Thus, for example, if a policy states that the service provider only collects personal information from registered users, the policy is analyzed from the perspective of a registered user. Also, if certain actions are dependent on the user's consent, opt-in, or opt-out, it is assumed that the user consented, opted in, or did not opt out, respectively. As it was our goal to make the analysis results intuitively comprehensible to casual users, which needs to be confirmed by user studies, we tried to avoid technical terms. In particular, the term "personal information" is identical to what is known in the privacy community as personally identifiable information (PII) (while "information" on its own also encompasses non-PII, e.g., user agent information).



A

🟥 Collection of Personal Info (such as e-mail address)

🟩 No Combination with Info from outside Companies

🟩 No Advertising Tracking (e.g., no use of Ad Cookies)

🟩 No Disclosure of Personal Info to Advertisers

🟨 Personal Info is Archived for Unlimited Time

🟨 Stored and Transmitted Info is not Encrypted

Meaning of Grades and Symbols

A = Above Average Overall Privacy	🟩 = Good Privacy Practice
B = Average Overall Privacy	🟨 = Neutral Privacy Practice
C = Below Average Overall Privacy	🟥 = Bad Privacy Practice

Learn More

Figure 3: Privee extension screenshot and detailed label view. The result of the privacy policy analysis is shown to the user in a pop-up.

It is noteworthy that some of the analyzed criteria correspond to the semantics of the P3P Compact Specification [2]. For example, the P3P token NOI indicates that a Web service does not collect identified data while ALL means that it has access to all identified data. Thus, NOI and ALL correspond to our collection category. Also, in P3P the token IND means that information is retained for an indeterminate period of time, and, consequently, is equivalently expressed when our classifier comes to the conclusion that no limited retention exists. Further, PSA, PSD, IVA, and IVD are tokens similar to our profiling category. Generally, the correspondence between the semantics of the P3P tokens and our categories suggests that it is possible to automatically classify natural language privacy policies to obtain the same information that Web services would include in P3P policies without actually requiring them to have such.

4.7 Labeler

Our extension's labeler is responsible for creating an output label. As it was shown that users casually familiar with privacy questions were able to understand privacy policies faster and more accurately when those policies were presented in a standardized format [49] and that most users had a preference for standardized labels over full policy texts [49, 50], we created a short standardized label format. Generally, a label can be structured in one or multiple dimensions. The multidimensional approach has the advantage that it can succinctly display different privacy practices for different types of information. However, we chose a one-dimensional format as such were shown to be substantially more comprehensible [51, 66].

In addition to the descriptions for the classifications, the labeler also labels each policy with an overall letter grade, which depends on the classifications. More specifically, the grade is determined by the number of points, p, a policy is assigned. For collection, profiling, ad tracking, and ad disclosure a policy receives one minus point, respectively. However, for not allowing one of these practices a policy receives one plus point. However, a policy receives a plus point for featuring limited retention or encryption, respectively. As most policies in the training set had zero points, we took zero points as a mean and assigned grades as follows:

- A (above average overall privacy) if $p > 1$;
- B (average overall privacy) if $1 \leq p \geq -1$;
- C (below average overall privacy) if $p < -1$.

After the points are assigned to a policy, the corresponding label is displayed to the user as shown in Figure 3. As we intended to avoid confusion about the meaning of icons [45], we used short descriptions instead. The text in the pop-up is animated. If the user moves the mouse over it, further information is provided. The user can also find more detailed explanations about the categories and the grading by clicking on the blue ”Learn More” link at the bottom of the label. It should be noted that analysis results retrieved from ToS;DR usually differ in content from our classification results, and are, consequently, displayed in a different label format.

5 Experimental Results

For our experiments we ran our Privee extension on a test set of 50 policies. Before this test phase we trained the ML classifier (with the 100 training policies that are included in the extension package) and tuned it (with a validation set of 50 policies). During the training, validation, and test phases we disabled the retrieval of crowdsourcing results. Consequently, our experimental results only refer to rule and ML classification. The policies of the test and validation sets were selected according to the same criteria as described for the training set in Section

	Base.	Acc.	Prec.	Rec.	F-1
Overall	**68%**	**84%**	**94%**	**89%**	**90%**
Collection	100%	100%	100%	100%	100%
Encryption	52%	98%	96%	100%	98%
Ad Tracking	64%	96%	94%	100%	97%
L. Retention	74%	90%	83%	77%	80%
Profiling	52%	86%	100%	71%	83%
Ad Disclosure	66%	76%	69%	53%	60%

Table 1: Privee extension performance overall and per category. For the 300 test classifications (six classifications for each of the 50 test policies) we observed 27 misclassifications. 154 classifications were made by the rule classifier and 146 by the ML classifier. The rule classifier had 11 misclassifications (2 false positives and 9 false negatives) and the ML classifier had 16 misclassifications (7 false positives and 9 false negatives). It may be possible to decrease the number of false negatives by adding more rules and training examples. For the ad tracking category the rule classifier had an F-1 score of 98% and the ML classifier had an F-1 score of 94%. For the profiling category the rule classifier had an F-1 score of 100% and the ML classifier had an F-1 score of 53%. 28% of the policies received a grade of A, 50% a B, and 22% a C.

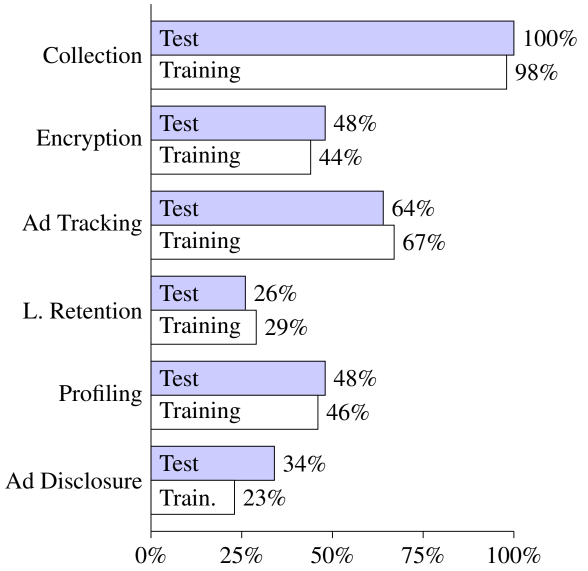


Figure 4: Annotation of positive cases in percent for the 50 test policies (blue) and the 100 training policies (white).

4.5. In this Section we first discuss the classification performance (Section 5.1), then the gold standard that we used to measure the performance (Section 5.2), and finally the computational performance (Section 5.3).

5.1 Classification Performance

In the validation phase we experimented with different classifier configurations for each of our six classification tasks. For the ad tracking and profiling categories the combination of the rule and ML classifier lead to the best results. However, for collection, limited retention, and ad disclosure the ML classifier on its own was preferable. Conversely, for the encryption category the rule classifier on its own was the best. It seems that the language used for describing encryption practices is often very specific making the rule classifier the first choice. Words such as “ssl” are very distinctive identifiers for encryption provisions. Other categories use more general language that could be used in many contexts. For example, phrases related to time periods must not necessarily refer to limited retention. For those instances the ML classifier seems to perform better. However, if categories exhibit both specific and general language the combination of the rule and ML classifier is preferable.

The results of our extension’s privacy policy analysis are based on the processing of natural language. However, as natural language is often subject to different interpretations, the question becomes how the results can be verified in a meaningful way. Commonly applied metrics for verifying natural language classification tasks are accuracy (Acc.), precision (Prec.), recall (Rec.), and F-1 score (F-1). Accuracy is the fraction of classifications that are correct [56]. Precision is the fraction of retrieved documents that are relevant, and recall is the fraction of relevant documents that are retrieved [56]. Precision and recall are often combined in their harmonic mean, known as the F-1 score [46].

In order to analyze our extension’s performance we calculated the accuracy, precision, recall, and F-1 score for the test policy set classifications. Table 1 shows the overall performance and the performance for each classification category. We also calculated the baseline accuracy (Base.) for comparison against the actual accuracy. The baseline accuracy for each category was determined by always selecting the classification corresponding to the annotation that occurred the most in the training set annotations, which we report in Figure 4. The baseline accuracy for the overall performance is the mean of the category baseline accuracies. Because the classification of privacy policies is a multi-label classification task, as described in Section 4.6.1, we calculated the overall results based on the method for measuring multi-label classifications given by Godbole and Sarawagi [42]. According to their method, for each document, d_j in set D, let t_j be the true set of labels and s_j be the predicted set of labels. Then we obtain the means by

$$Acc(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|t_j \cap s_j|}{|t_j \cup s_j|}, \tag{3}$$

$$Prec(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|t_j \cap s_j|}{|s_j|}, \tag{4}$$

$$Rec(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|t_j \cap s_j|}{|t_j|}, \tag{5}$$

$$F\text{-}1(D) = \frac{1}{|D|}\sum_{i=1}^{|D|} \frac{2\,Prec(d_j)\,Rec(d_j)}{(Prec(d_j)+Rec(d_j))}. \qquad (6)$$

From Table 1 it can be observed that the accuracies are at least as good as the corresponding baseline accuracies. For example, in the case of limited retention the baseline classifies all policies as not providing for limited retention because, as show in Figure 4, only 29% of the training policies were annotated as having a limited retention period, which would lead to a less accurate classification of 74% in the test set compared to the actual accuracy of 90%. For the collection category it should be noted that there is a strong bias because nearly every policy allows the collection of personal information. However, in our validation set we had two policies that did not allow this practice, but still were correctly classified by our extension. Generally, our F-1 performance results fall squarely within the range reported in the earlier works. For identifying law enforcement disclosures Ammar et Al. [16] achieved an F-1 score of 76% and Costante et al. reported a score of 83% for recognizing types of collected information [26] and 92% for identifying topics discussed in privacy policies [27].

In order to investigate the reasons behind our extension's performance we used two binary logistic regression models. Binary logistic regression is a statistical method for evaluating the dependence of a binary variable (the dependent variable) on one or more other variables (the independent variable(s)). In our first model each of the 50 test policies was represented by one data point with the dependent variable identifying whether it had any misclassification and the independent variables identifying (1) the policy's length in words, (2) its mean Semantic Diversity (SemD) value [44], and (3) whether there was any disagreement among the annotators in annotating the policy (Disag.). In our second model we represented each of 185 individual test classifications by one data point with the dependent variable identifying whether it was a misclassification and the independent variables identifying (1) the length (in words) of the text that the rule classifier or ML preprocessor extracted for the classification, (2) the text's mean SemD value, and (3) whether there was annotator disagreement on the annotation corresponding to the classification.

Hoffman et al.'s [44] SemD value is an ambiguity measure for words based on latent semantic analysis, that is, the similarity of contexts in which words are used. It can range from 0 (highly unambiguous) to 2.5 (highly ambiguous). We represented the semantic diversity of a document (i.e., a policy or extracted text) by the mean SemD value of its words. However, as Hoffman et al. only provide SemD values for words on which they had sufficient analytical data (31,739 different words in total), some words could not be taken into account for calculating a document's mean SemD value. Thus, in order to avoid skewing of mean SemD values in our models, we only considered documents that had SemD values for at least 80% of their words. In our first model all test policies were above this threshold. However, in our second model we excluded some of the 300 classifications. Particularly, all encryption classifications were excluded because words, such as "encryption" and "ssl" occurred often and had no SemD value. Also, in the second model the mean SemD value of an extracted text was calculated after stemming its words with the Porter stemmer and obtaining the SemD values for the resulting word stems (while the SemD value of each word stem was calculated from the mean SemD value of all words that have the respective word stem).

Per Policy	Length	SemD	Disag.
Mean	2873.4	2.08	0.6
Significance (P)	0.64	0.74	0.34
Odds Ratio (Z)	1.15	1.11	0.54
95% Confidence Interval (Z)	0.64-2.08	0.61-2.01	0.16-1.89

Table 2: Results of the first logistic regression model. The Nagelkerke pseudo R^2 is 0.03 and the Hosmer and Lemeshow value 0.13.

Per Extr. Text	Length	SemD	Disag.
Mean	37.38	1.87	0.17
Significance (P)	0.22	**0.02**	0.81
Odds Ratio (Z)	0.58	**2.07**	0.86
95% Confidence Interval (Z)	0.24-1.38	1.12-3.81	0.25-2.97

Table 3: Results of the second logistic regression model. The Nagelkerke pseudo R^2 is 0.11 and the Hosmer and Lemeshow value 0.051.

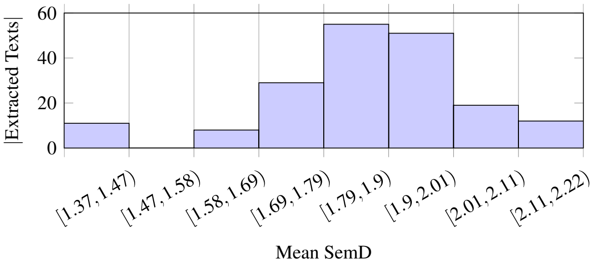


Figure 5: Mean SemD value distribution for the 185 extracted texts. The standard deviation is 0.17.

For our first model the results of our analysis are shown in Table 2 and for our second model in Table 3. Figure 5 shows the distribution of mean SemD values for the extracted texts in our second model. Using the

Wald test, we evaluated the relationship between an independent variable and the dependent variable through the P value relating to the coefficient of that independent variable. If the P value is less than 0.05, we reject the null hypothesis, i.e., that that coefficient is zero. Looking at our results, it is noteworthy that both models do not reveal a statistically relevant correlation between the annotator disagreements and misclassifications. Thus, a document with a disagreement did not have a higher likelihood of being misclassified than one without. However, it is striking that the second model has a P value of 0.02 for the SemD variable. Standardizing our data points into Z scores and calculating the odds ratios it becomes clear that an increase of the mean SemD value in an extracted text by 0.17 (one standard deviation) increased the likelihood of a misclassification by 2.07 times (odds ratio). Consequently, our second model shows that the ambiguity of text in privacy policies, as measured by semantic diversity, has statistical significance for whether a classification decision is more likely to succeed or fail.

Besides evaluating the statistical significance of individual variables, we also assessed the overall model fit. While the goodness of fit of linear regression models is usually evaluated based on the R^2 value, which measures the square of the sample correlation coefficient between the actual values of the dependent variable and the predicted values (in other words, the R^2 value can be understood as the proportion of the variance in a dependent variable attributable to the variance in the independent variable), there is no consensus for measuring the fit of binary logistic regression models. Various pseudo R^2 metrics are discussed. We used the Nagelkerke pseudo R^2 because it can range from 0 to 1 allowing an easy comparison to the regular R^2 (which, however, has to account for the fact that the Nagelkerke pseudo R^2 is often substantially lower than the regular R^2). While the Nagelkerke pseudo R^2 of 0.03 for our first model indicates a poor fit, the value of 0.11 for our second model can be interpreted as moderate. Further, the Hosmer and Lemeshow test, whose values were over 0.05 for both of our models, demonstrates the model fit as well.

In addition to the experiments just discussed, we also evaluated our models with further independent variables. Specifically, we evaluated our first model with the policy publication year, the second model with the extracted texts' mean tf-idf values, and both models with Flesch-Kincaid readability scores as independent variables. Also, using only ML classifications we evaluated our second model with the number of available training examples as independent variable. Only for the latter we found statistical significance at the 0.05 level. The number of training examples correlated to ML classification performance, which confirms Ammar et al.'s respective conjecture [16]. The more training examples the ML classifier had, the less likely a misclassification became.

5.2 Inter-annotator Agreement

Having discussed the classification performance, we now turn to the gold standard that we used to measure that performance. For our performance results to be reliable our gold standard must be reliable. One way of producing a gold standard for privacy policies is to ask the providers whose policies are analyzed to explain their meaning [11]. However, this approach should not be used, at least in the U.S., because the Restatement of Contracts provides that a contract term is generally given the meaning that *all* parties associate with it (Restatement (Second) of Contracts, §201). Consequently, policies should be interpreted from the perspective of both the provider and user. The interpretation would evaluate whether their perspectives lead to identical meanings or, if that is not the case, which one should prevail under applicable principles of legal interpretation. In addition, since technical terms are generally given technical meaning (Restatement (Second) of Contracts, §202(3)(b)), it would be advantageous if the interpretation is performed by annotators familiar with the terminology commonly used in privacy policies. The higher the number of annotations on which the annotators agree, that is, the higher the inter-annotator agreement, the more reliable the gold standard will be.

Because the annotation of a large number of documents can be very laborious, it is sufficient under current best practices for producing a gold standard to measure inter-annotator agreement only on a data sample [62], such that it can be inferred that the annotation of the remainder documents is reliable as well. Following this practice, we only measured the inter-annotator agreement for our test set, which would then provide an indicator for the reliability of our training and validation set annotation as well. To that end, one author annotated all policies and additional annotations were obtained for the test policies from two other annotators. All annotators worked independently from each other. As the author who annotated the policies studied law and has expertise in privacy law and the two other annotators were law students with training in privacy law, all annotators were considered equally qualified, and the annotations for the gold standard were selected according to majority vote (i.e., at least two annotators agreed). After the annotations of the test policies were made, we ran our extension on these policies and compared its classifications to the annotations, which gave us the results in Table 1.

The reliability of our gold standard depends on the degree to which the annotators agreed on the annotations. There are various measures for inter-annotator agreement. One basic measure is the count of disagreements.

	Disag.	% Ag.	K.'s α/F.'s κ
Overall	**8.12**	**84%**	**0.77**
Collection	0	100%	1
Encryption	6	88%	0.84
Ad Tracking	7	86%	0.8
L. Retention	9	82%	0.68
Profiling	11	78%	0.71
Ad Disclosure	16	68%	0.56

Table 4: Inter-annotator agreement for the 50 test policies. The values for Krippendorff's α and Fleiss' κ are identical.

Per Policy	Length	SemD	Flesch-K.
Mean	2873.4	2.08	14.53
Significance (P)	0.2	0.11	0.76
Odds Ratio (Z)	1.65	1.87	1.12
95% Confidence Interval (Z)	0.78-3.52	0.87-4	0.55-2.29

Table 5: Results of the third logistic regression model. The Nagelkerke pseudo R^2 is 0.19 and the Hosmer and Lemeshow value 0.52.

Per Section	Length	SemD	Flesch-K.
Mean	306.76	2.08	15.59
Significance (P)	0.29	**0.04**	0.49
Odds Ratio (Z)	1.18	**1.51**	0.86
95% Confidence Interval (Z)	0.87-1.6	1.02-2.22	0.56-1.32

Table 6: Results of the fourth logistic regression model. The Nagelkerke pseudo R^2 is 0.05 and the Hosmer and Lemeshow value 0.83.

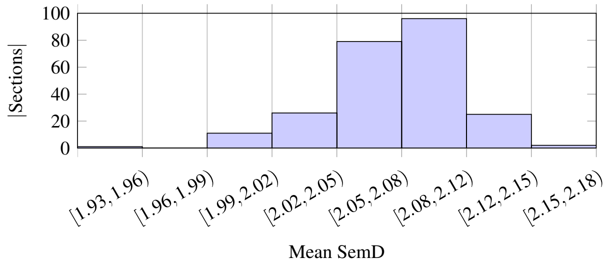


Figure 6: Mean SemD value distribution for the 240 policy sections. The standard deviation is 0.03.

Another one is the percentage of agreement (% Ag.), which is the fraction of documents on which the annotators agree [17]. However, disagreement count and percentage of agreement have the disadvantage that they do not account for chance agreement. In this regard, chance-corrected measures, such as Krippendorff's α (K.'s α) [53] and Fleiss' κ (F.'s κ) [40] are superior. For Krippendorff's α and Fleiss' κ the possible values are constrained to the interval $[-1;1]$, where 1 means perfect agreement, -1 means perfect disagreement, and 0 means that agreement is equal to chance [37]. Generally, values above 0.8 are considered as good agreement, values between 0.67 and 0.8 as fair agreement, and values below 0.67 as dubious [56]. However, those ranges are only guidelines [17]. Particularly, ML algorithms can tolerate data with lower reliability as long as the disagreement looks like random noise [68].

Based on the best practices and guidelines for interpreting inter-annotator agreement measurements, our results in Table 4 confirm the general reliability of our annotations and, consequently, of our gold standard. For every individual category, except for the ad disclosure category, we obtained Krippendorff's α values indicating fair or good agreement. In addition, the overall mean agreement across categories is 0.77, and, therefore, provides evidence for fair overall agreement as well. For the overall agreement it should be noted that, corresponding to the multi-label classification task, the annotation of privacy policies is a multi-label annotation task as well. However, there are only very few multi-label annotation metrics, such as Passonneau's Measuring Agreement on Set-valued Items (MASI) [61]. As none of the metrics were suitable for our purposes, we selected as overall metric the mean over the results of the individual classification categories.

We investigated our inter-annotator agreement results by applying a third and fourth binary logistic regression model. In our third model each of the 50 test policies was represented by one data point with the dependent variable identifying whether the annotators had any disagreement in annotating the policy and the independent variables identifying (1) the policy's length in words, (2) its mean SemD value, and (3) its Flesch-Kincaid score. In our fourth model we represented each of 240 individual annotations by one data point with the dependent variable identifying whether the annotators disagreed for that annotation and the independent variables identifying (1) the length (in words) of the policy text section that the annotation is referring to, (2) the section's mean SemD value, and (3) its Flesch-Kincaid score. For the fourth model we excluded some of the 300 annotations because not every policy had a section for each category. For example, some policies did not discuss advertisement or disclosure of information. The Flesch-Kincaid readability score measures the number of school years an average reader would need to understand a text.

For our third and fourth model the results of our analysis are shown in Table 5 and 6, respectively. Figure 6 shows the distribution of mean SemD values for the policy sections in our fourth model. Both models were significant, as indicated by their Nagelkerke and Hosmer and Lemeshow values. Our results confirm that the readability of policies, as measured by the Flesch-Kincaid score, does not impact their comprehensibility [58]. In our third model we were unable to identify any statistically relevant variables (although, semantic diversity and length may be statistically significant in a larger data set). However, our fourth model proved to be more meaningful. Remarkably, corresponding to our finding in Section 5.1, according to which classifier performance correlates to semantic diversity, the statistically relevant P value of 0.04 for the mean SemD variable also indicates a correlation of inter-annotator agreement to semantic diversity. Standardizing our data points into Z scores and calculating the odds ratios it becomes clear that an increase of the mean SemD value of a section by 0.03 (one standard deviation) increased the likelihood of a disagreement by 1.51 times (odds ratio). It is astounding that even qualified annotators trained in privacy law had difficulties to avoid disagreements when semantic diversity increased to slightly above-mean levels.

While neither our first nor our second model in Section 5.1 showed a correlation between inter-annotator agreement and classifier performance, the results for our second and fourth model demonstrate that performance and agreement both correlate to one common variable—semantic diversity. More specifically, performance correlates to the semantic diversity of extracted text phrases and agreement correlates to the semantic diversity of policy sections. This result suggests, for example, that the relatively high number of misclassifications and disagreements in the ad disclosure category is inherent in the nature of the category. Indeed, in cases of fuzzy categories disagreements among annotators do not necessarily reflect a quality problem of the gold standard, but rather a structural property of the annotation task, which can serve as an important source of empirical information about the structural properties of the investigated category [13]. Thus, it is no surprise that for all six categories the values of Krippendorff's α correlate to the F-1 scores. The higher the value of Krippendorff's α, the higher the F-1 score. Figure 7 shows the correlation.

As both classifier performance and inter-annotator agreement decrease with an increase in semantic diversity, the practicability of the notice-and-choice principle becomes questionable. After all, privacy policies can only provide adequate notice (and choice) if they are not too ambiguous. In order to further examine policy ambiguity we calculated the mean SemD value for our test policies over time. Our test set analysis exhibited a statistically significant trend of decreasing semantic diversity with a P value of 0.049. Figure 8 illustrates our approach. We can think of two explanations for the decrease over time. First, it could be a consequence of the FTC's enforcement actions and its call for policies to "be clearer, shorter, and more standardized" [38]. Second, we might be in the midst of a consolidation process leading to more standardized policy language. As de Maat et al [34] observed, drafters of legal documents tend to use language that adheres to writing conventions of earlier texts and similar statements. Independent of the reason, our result suggests that the notice-and-choice principle can overcome the problem of ambiguity over time.

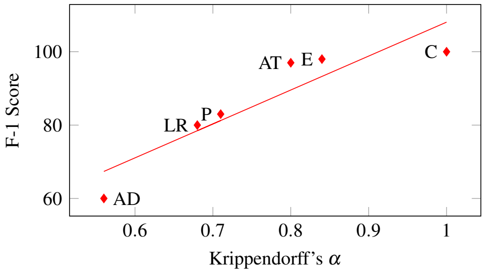


Figure 7: Linear regression plot with the F-1 score as dependent variable and Krippendorff's α as independent variable. The coordinate labels identify the categories: AD = Ad Disclosure, LR = Limited Retention, P = Profiling, AT = Ad Tracking, E = Encryption, and C = Collection. With an R^2 value of 0.83 the model has an excellent fit, which, however, should be interpreted in light of the small number of data points.

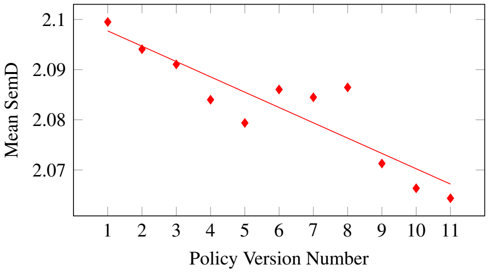


Figure 8: Linear regression plot for Symantec's privacy policy (which was part of our test set) with the mean SemD value of a policy version as dependent variable and the policy version number as independent variable. The first version of Symantec's policy [73] dates back to August 5, 1999, and the eleventh version [72] was adopted on August 12, 2013. The mean SemD value of Symantec's privacy policy decreased from 2.1 in the first version to 2.06 in the eleventh version as shown. We observed a similar decrease for 29 out of 44 test policies (6 of the test policies were only available in a single version and, therefore, could not be included in our analysis. However, for the 44 included policies we obtained on average 8 different versions over time.).

5.3 Computational Performance

We finish the discussion of our experimental results with our extension's computational performance. We report the mean duration in seconds for obtaining analysis results for each of 50 randomly selected policies from ToS;DR (Crowdsourcing), processing each of the 50 test policies (Classifier), and processing each of the 50 test

Per Policy	Crowdsourcing	Classifier	Training
Mean	0.39 sec	0.78 sec	20.29 sec

Table 7: Computational performance of the Privee extension. The performance was evaluated on a Windows laptop with Intel Core2 Duo CPU at 2.13 GHz with 4 GB RAM. The space requirements for the installation on the hard disk are 2.11 MB (including 1.7 MB of training data and 286 KB for the jQuery library) and additional 230 KB during the program execution for storing training results.

policies each with initial training (Training) in Table 7. Notably, retrieving policy results from ToS;DR is twice as fast as analyzing a policy with our classifiers.

6 Conclusion

We introduced Privee—a novel concept for analyzing natural language privacy policies based on crowdsourcing and automatic classification techniques. We implemented Privee in a proof of concept browser extension for Google Chrome, and our automatic classifiers achieved an overall F-1 score of 90%. Our experimental results revealed that the automatic classification of privacy policies encounters the same constraint as human interpretation—the ambiguity of natural language, as measured by semantic diversity. Such ambiguity seems to present an inherent limitation of what automatic privacy policy analysis can accomplish. Thus, on a more fundamental level, the viability of the notice-and-choice principle might be called into question altogether. However, based on the decrease of policy ambiguity over time we would caution to draw such conclusion. We remain optimistic that the current notice-and-choice ecosystem is workable and can be successfully supplemented by Privee.

The most important task for making the notice-and-choice principle work is to decrease policy ambiguity. However, other areas require work as well: What are the types of information that policies should be analyzed for? What is the most usable design? What are the best features and algorithms? Are more intricate ML or natural language processing algorithms better at resolving ambiguities? What is the ideal size and composition of the training set? How can the interaction between the classifier and crowdsourcing analysis be improved? In particular, how can a program connect to many crowdsourcing repositories, and, possibly, decide which analysis is the best? Can crowdsourced policy results be used by the classifiers as training data? How can it be assured that the crowdsourcing results are always up to date? How can the quality and consistency of crowdsourcing and ML analyses be guaranteed? And, finally, what solutions are viable for different legal systems and the mobile world?

7 Acknowledgments

For their advice and help we thank Kapil Thadani, Rebecca Passonneau, Florencia Marotta-Wurgler, Sydney Archibald, Shaina Hyder, Jie S. Li, Gonzalo Mena, Paul Schwartz, Kathleen McKeown, Tony Jebara, Jeroen Geertzen, Michele Nelson, and the commentators at the 7th Annual Privacy Law Scholars Conference (2014) and the Workshop on the Future of Privacy Notice and Choice (2014). We also thank the anonymous reviewers for their valuable feedback.

8 Availability

Our Privee extension is available at `http://www.sebastianzimmeck.de/publications.html`.

References

[1] Claridge v. RockYou, Inc., 785 F. Supp. 2d 855 (N.D. Cal. 2011).

[2] P3P compact policy cross-reference. http://compactprivacypolicy.org/compact_token_reference.htm. Last accessed: July 1, 2014.

[3] Privacy Icons. http://www.azarask.in/blog/post/privacy-icons/. Last accessed: July 1, 2014.

[4] privacychoice. http://www.privacychoice.org. Last accessed: July 1, 2014.

[5] Terms of Service; Didn't Read (ToS;DR). http://tosdr.org/index.html. Last accessed: July 1, 2014.

[6] Top sites in United States. http://www.alexa.com/topsites/countries/US. Last accessed: July 1, 2014.

[7] TOSBack. http://tosback.org/. Last accessed: July 1, 2014.

[8] TOSBack2. https://github.com/pde/tosback2. Last accessed: July 1, 2014.

[9] Usable Privacy Policy Project. http://www.usableprivacy.org/home. Last accessed: July 1, 2014.

[10] Platform for Internet Content Selection (PICS). http://www.w3.org/PICS/, 1997. Last accessed: July 1, 2014.

[11] KnowPrivacy. http://knowprivacy.org/, June 2009. Last accessed: July 1, 2014.

[12] eXtensible Access Control Markup Language (XACML) version 3.0. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html, Jan. 2013. Last accessed: July 1, 2014.

[13] Adreevskaia, A., and Bergler, S. Mining WordNet for fuzzy sentiment: Sentiment tag extraction from WordNet glosses. In *11th conference of the European chapter of the Association for Computational Linguistics* (Stroudsburg, PA, USA, 2006), EACL '06, ACL, pp. 209–216.

[14] Agrawal, R., Kiernan, J., Srikant, R., and Xu, Y. An XPath-based preference language for P3P. In *Proceedings of the 12th international conference on World Wide Web* (New York, NY, USA, 2003), WWW '03, ACM, pp. 629–639.

[15] Aïmeur, E., Gambs, S., and Ho, A. UPP: User privacy policy for social networking sites. In *Fourth International Conference on Internet and Web applications and services* (Washington, DC, USA, 2009), ICIW '09, IEEE Computer Society, pp. 267–272.

[16] Ammar, W., Wilson, S., Sadeh, N., and Smith, N. Automatic categorization of privacy policies: A pilot study. Tech. Rep. CMU-ISR-12-114, CMU-LTI-12-019, Carnegie Mellon University, Dec. 2012.

[17] Artstein, R., and Poesio, M. Inter-coder agreement for computational linguistics. *Comput. Linguist. 34*, 4 (Dec. 2008), 555–596.

[18] Ashley, P., Hada, S., Karjoth, G., Powers, C., and Schunter, M. Enterprise Privacy Authorization Language (EPAL 1.2). Tech. rep., IBM, Nov. 2003.

[19] Biagioli, C., Francesconi, E., Passerini, A., Montemagni, S., and Soria, C. Automatic semantics extraction in law documents. In *Proceedings of the 10th International Conference on Artificial Intelligence and Law* (New York, NY, USA, 2005), ICAIL '05, ACM, pp. 133–140.

[20] Breaux, T. D., and Antón, A. I. Mining rule semantics to understand legislative compliance. In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society* (New York, NY, USA, 2005), WPES '05, ACM, pp. 51–54.

[21] Breaux, T. D., and Antón, A. I. Analyzing regulatory rules for privacy and security requirements. *IEEE Trans. Software Eng. 34*, 1 (Jan. 2008), 5–20.

[22] Brodie, C., Karat, C.-M., Karat, J., and Feng, J. Usable security and privacy: a case study of developing privacy management tools. In *Proceedings of the 2005 Symposium On Usable Privacy and Security* (New York, NY, USA, 2005), SOUPS '05, ACM, pp. 35–43.

[23] Brodie, C. A., Karat, C.-M., and Karat, J. An empirical study of natural language parsing of privacy policy rules using the SPARCLE policy workbench. In *Proceedings of the second Symposium On Usable Privacy and Security* (New York, NY, USA, 2006), SOUPS '06, ACM, pp. 8–19.

[24] Byers, S., Cranor, L. F., Kormann, D., and McDaniel, P. Searching for privacy: design and implementation of a P3P-enabled search engine. In *Proceedings of the 4th international conference on Privacy Enhancing Technologies* (Berlin, Heidelberg, Germany, 2005), PET '04, Springer, pp. 314–328.

[25] Ciocchetti, C. A. The future of privacy policies: A privacy nutrition label filled with fair information practices. *J. Marshall J. Computer & Info. L. 26* (2008), 1–46.

[26] Costante, E., den Hartog, J., and Petkovic, M. What websites know about you. In *DPM/SETOP* (Berlin, Heidelberg, Germany, 2012), R. D. Pietro, J. Herranz, E. Damiani, and R. State, Eds., vol. 7731 of *Lecture Notes in Computer Science*, Springer, pp. 146–159.

[27] Costante, E., Sun, Y., Petković, M., and den Hartog, J. A machine learning solution to assess privacy policy completeness: (short paper). In *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society* (New York, NY, USA, 2012), WPES '12, ACM, pp. 91–96.

[28] Cranor, L. F. Necessary but not sufficient: Standardized mechanisms for privacy notice and choice. *J. on Telecomm. and High Tech. L. 10*, 2 (2012), 273–307.

[29] Cranor, L. F., Dobbs, B., Egelman, S., Hogben, G., Humphrey, J., Langheinrich, M., Marchiori, M., Presler-Marshall, M., Reagle, J. M., Schunter, M., Stampley, D. A., and Wenning, R. The Platform for Privacy Preferences 1.1 (P3P1.1) specification. World Wide Web Consortium, Note NOTE-P3P11-20061113, November 2006.

[30] Cranor, L. F., Guduru, P., and Arjula, M. User interfaces for privacy agents. *ACM Trans. Comput.-Hum. Interact. 13*, 2 (June 2006), 135–178.

[31] Cranor, L. F., Langheinrich, M., and Marchiori, M. A P3P Preference Exchange Language 1.0 (APPEL 1.0). World Wide Web Consortium, Working Draft WD-P3P-preferences-20020415, April 2002.

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

[33] Cranor, L. F., and Reidenberg, J. R. Can user agents accurately represent privacy notices? *TPRC* (Sept. 2002).

[34] de Maat, E., Krabben, K., and Winkels, R. Machine learning versus knowledge based classification of legal texts. In *Proceedings of the 2010 conference on Legal Knowledge and Information Systems: JURIX 2010: The Twenty-Third Annual Conference* (Amsterdam, The Netherlands, The Netherlands, 2010), IOS Press, pp. 87–96.

[35] de Maat, E., and Winkels, R. Automatic classification of sentences in dutch laws. In *Proceedings of the 2008 conference on Legal Knowledge and Information Systems: JURIX 2008: The Twenty-First Annual Conference* (Amsterdam, The Netherlands, The Netherlands, 2008), IOS Press, pp. 207–216.

[36] de Maat, E., and Winkels, R. A next step towards automated modelling of sources of law. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law* (New York, NY, USA, 2009), ICAIL '09, ACM, pp. 31–39.

[37] Di Eugenio, B., and Glass, M. The kappa statistic: a second look. *Comput. Linguist. 30*, 1 (Mar. 2004), 95–101.

[38] Federal Trade Commission. Protecting consumer privacy in an era of rapid change. `http://www.ftc.gov/reports/protecting-consumer-privacy-era-rapid-change-recommendations-businesses-policymakers`, Mar. 2012. Last accessed: July 1, 2014.

[39] Fischer-Hübner, S., and Zwingelberg, H. UI prototypes: Policy administration and presentation - version 2. Tech. Rep. D4.3.2, Karlstad University, 2010.

[40] Fleiss, J. Measuring nominal scale agreement among many raters. *Psychological Bulletin 76*, 5 (1971), 378–382.

[41] Francesconi, E., and Passerini, A. Automatic classification of provisions in legislative texts. *Artif. Intell. Law 15*, 1 (Mar. 2007), 1–17.

[42] Godbole, S., and Sarawagi, S. Discriminative methods for multi-labeled classification. In *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining* (Berlin, Heidelberg, Germany, 2004), Springer, pp. 22–30.

[43] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. The WEKA data mining software: An update. *SIGKDD Explor. Newsl. 11*, 1 (Nov. 2009), 10–18.

[44] Hoffman, P., Ralph, M. L., and Rogers, T. Semantic diversity: A measure of semantic ambiguity based on variability in the contextual usage of words. *BRM 45*, 3 (2013), 718–730.

[45] Holtz, L.-E., Nocun, K., and Hansen, M. Towards displaying privacy information with icons. In *Privacy and Identity Management for Life* (Berlin, Heidelberg, Germany, 2011), S. Fischer Hübner, P. Duquenoy, M. Hansen, R. Leenes, and G. Zhang, Eds., vol. 352 of *IFIP Advances in Information and Communication Technology*, Springer, pp. 338–348.

[46] Hripcsak, G., and Rothschild, A. S. Technical brief: Agreement, the F-measure, and reliability in information retrieval. *JAMIA 12*, 3 (2005), 296–298.

[47] KARJOTH, G., SCHUNTER, M., AND WAIDNER, M. Platform for Enterprise Privacy Practices: privacy-enabled management of customer data. In *Proceedings of the 2nd international conference on Privacy Enhancing Technologies* (Berlin, Heidelberg, Germany, 2003), PET '02, Springer, pp. 69–84.

[48] KELLEY, P. G. Designing a privacy label: assisting consumer understanding of online privacy practices. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2009), CHI EA '09, ACM, pp. 3347–3352.

[49] KELLEY, P. G., BRESEE, J., CRANOR, L. F., AND REEDER, R. W. A "nutrition label" for privacy. In *Proceedings of the 5th Symposium On Usable Privacy and Security* (New York, NY, USA, 2009), SOUPS '09, ACM, pp. 4:1–4:12.

[50] KELLEY, P. G., CESCA, L., BRESEE, J., AND CRANOR, L. F. Standardizing privacy notices: an online study of the nutrition label approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2010), CHI '10, ACM, pp. 1573–1582.

[51] KELLEY, P. G., CRANOR, L. F., AND SADEH, N. Privacy as part of the app decision-making process. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2013), CHI '13, ACM, pp. 3393–3402.

[52] KOBYLIŃSKI, L., AND PRZEPIÓRKOWSKI, A. Definition extraction with balanced random forests. In *Proceedings of the 6th international conference on Advances in Natural Language Processing* (Berlin, Heidelberg, Germany, 2008), GoTAL '08, Springer, pp. 237–247.

[53] KRIPPENDORFF, K. *Content analysis: An introduction to its methodology.* SAGE, Beverly Hills, CA, USA, 1980.

[54] KUHN, F. A description language for content zones of German court decisions. In *Proceedings of the LREC 2010 Workshop on the Semantic Processing of Legal Texts* (2010), SPLeT '10, pp. 1–7.

[55] LEON, P. G., CRANOR, L. F., MCDONALD, A. M., AND MCGUIRE, R. Token attempt: The misrepresentation of website privacy policies through the misuse of P3P compact policy tokens. In *Proceedings of the 9th Annual ACM Workshop on Privacy in the Electronic Society* (New York, NY, USA, 2010), WPES '10, ACM, pp. 93–104.

[56] MANNING, C. D., RAGHAVAN, P., AND SCHÜTZE, H. *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA, 2008.

[57] MAROTTA-WURGLER, F. Does contract disclosure matter? *JITE 168*, 1 (2012), 94–119.

[58] MCDONALD, A. M., REEDER, R. W., KELLEY, P. G., AND CRANOR, L. F. A comparative study of online privacy policies and formats. In *Proceedings of the 9th international symposium on Privacy Enhancing Technologies* (Berlin, Heidelberg, Germany, 2009), PETS '09, Springer, pp. 37–55.

[59] MOENS, M.-F., BOIY, E., PALAU, R. M., AND REED, C. Automatic detection of arguments in legal texts. In *Proceedings of the 11th International Conference on Artificial Intelligence and Law* (New York, NY, USA, 2007), ICAIL '07, ACM, pp. 225–230.

[60] NATIONAL TELECOMMUNICATIONS AND INFORMATION ADMINISTRATION. Short form notice code of conduct to promote transparency in mobile app practices. `http://www.ntia.doc.gov/files/ntia/publications/july_25_code_draft.pdf`, July 2013. Last accessed: July 1, 2014.

[61] PASSONNEAU, R. Measuring Agreement on Set-valued Items (MASI) for semantic and pragmatic annotation. In *Proceedings of the international Conference on Language Resources and Evaluation* (2006), LREC '06.

[62] PASSONNEAU, R. J., AND CARPENTER, B. The benefits of a model of annotation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse* (Stroudsburg, PA, USA, 2013), ACL, pp. 187–195.

[63] PORTER, M. An algorithm for suffix stripping. *Program: electronic library and information systems 14*, 3 (1980), 130–137.

[64] RAJARAMAN, A., AND ULLMAN, J. D. *Mining of massive datasets.* Cambridge University Press, New York, NY, USA, 2012.

[65] REEDER, R. W. *Expandable Grids: a user interface visualization technique and a policy semantics to support fast, accurate security and privacy policy authoring.* PhD thesis, Pittsburgh, PA, USA, 2008. AAI3321049.

[66] REEDER, R. W., KELLEY, P. G., MCDONALD, A. M., AND CRANOR, L. F. A user study of the Expandable Grid applied to P3P privacy policy visualization. In *Proceedings of the 7th ACM Workshop on Privacy in the Electronic Society* (New York, NY, USA, 2008), WPES '08, ACM, pp. 45–54.

[67] REIDENBERG, J. R. The use of technology to assure internet privacy: Adapting labels and filters for data protection. *Lex Electronica 3*, 2 (1997).

[68] REIDSMA, D., AND CARLETTA, J. Reliability measurement without limits. *Comput. Linguist. 34*, 3 (Sept. 2008), 319–326.

[69] RUBINSTEIN, I. S. Privacy and regulatory innovation: Moving beyond voluntary codes. *ISJLP 6*, 3 (2011), 355–423.

[70] STAMEY, J. W., AND ROSSI, R. A. Automatically identifying relations in privacy policies. In *27th ACM International Conference on Design of Communication* (New York, NY, USA, 2009), SIGDOC '09, ACM, pp. 233–238.

[71] STEDE, M., AND KUHN, F. Identifying the content zones of German court decisions. In *Lecture Notes in Business Information Processing* (Berlin, Heidelberg, Germany, 2009), vol. 37 of *BIS '09*, Springer, pp. 310–315.

[72] SYMANTEC CORPORATION. Complete privacy statement. `http://web.archive.org/web/20131028120625/http://www.symantec.com/about/profile/policies/privacy.jsp`. Last accessed: July 1, 2014.

[73] SYMANTEC CORPORATION. Global privacy statement for Symantec. `http://web.archive.org/web/19991012020231/http://symantec.com/legal/privacy.html`. Last accessed: July 1, 2014.

[74] TSOUMAKAS, G., AND KATAKIS, I. Multi label classification: An overview. *IJDWM 3*, 3 (2007), 1–13.

[75] WESTERHOUT, E. Definition extraction using linguistic and structural features. In *Proceedings of the 1st Workshop on Definition Extraction* (Stroudsburg, PA, USA, 2009), WDE '09, ACL, pp. 61–67.

[76] WESTERHOUT, E. Extraction of definitions using grammar-enhanced machine learning. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop* (Stroudsburg, PA, USA, 2009), EACL '09, ACL, pp. 88–96.

[77] WU, G., GREENE, D., AND CUNNINGHAM, P. Merging multiple criteria to identify suspicious reviews. In *Proceedings of the Fourth ACM Conference on Recommender Systems* (New York, NY, USA, 2010), RecSys '10, ACM, pp. 241–244.

[78] YANG, J., YESSENOV, K., AND SOLAR-LEZAMA, A. A language for automatically enforcing privacy policies. In *Proceedings of the 39th annual ACM SIGPLAN-SIGACT symposium on Principles Of Programming Languages* (New York, NY, USA, 2012), POPL '12, ACM, pp. 85–96.