Detection and Labeling of Personal Identifiable Information in E-mails

Christoph Bier, Jonas Prior

Detection and Labeling of Personal Identifiable Information in E-Mails

Christoph Bier and Jonas Prior

Fraunhofer Institute of Optronics, System Technologies and Image Exploitation
IOSB, Karlsruhe, Germany
{christoph.bier|jonas.prior}@iosb.fraunhofer.de

Abstract. The protection of personal identifiable information (PII) is increasingly demanded by customers and data protection regulation. To safeguard PII a organization has to find out which incoming communication actually contains it. Only then PII can be labeled, tracked, and protected. E-mails are one of the main means of communication. They consist of unstructured data difficult to classify. We developed an automated detection system for PII in e-mails and connected it to a usage control infrastructure. Our concept is based on previous findings in the area of spam detection. We tested our approach with a data set in a customer service scenario. The evaluation shows that the utilization of Bayes-classification is very promising to detect PII.

1 Introduction

The European data protection regulation (95/46/EC) requires enterprises, e.g., telecommunication providers, to comply with the clients' privacy rights, to ensure the protection of personally identifiable information (PII) and to fulfill the right to information. The right to information entitles the data subject to request from the data controller information on origin, transfer, and purpose of processing of his PII at any time. Thus, the usage and processing of PII needs to be safeguarded and controlled by detection, labeling and tracking. With the tremendous growth of data, it is not feasible to do that ad hoc. As a solution, usage control and provenance tracking methods [1] are proposed in recent research. But one of the major and yet unsolved problems remains: How does one know which received or created data contains PII? How can privacy policies and PII come together? Up to now, this is mostly done manually. But it is not yet possible to detect unstructured PII and annotate them with policies automatically.

As one of the most frequently used means of communication between clients and companies, especially in customer service, e-mails offer a rich source of PII. Many of them contain highly confidential customer data like banking accounts, usage patterns or contractual data, while many others like internal e-mails and newsletters do not. Overall, PII in e-mails is heterogeneous and therefore difficult to identify.

In this paper we present a model that can recognize e-mails containing PII and annotate them with policies (section 2). Our idea is to apply spam filter technologies to the classification of PII in incoming e-mails.

We design an integrated model of PII detection, usage control and provenance tracking, that can attach and enforce policies for sensitive data in order to protect them. A prototype is developed as part of a mail user agent to show the general feasibility of our idea (section 3). An extensive evaluation based on realistic test data shows that our approach is reasonably justified and promising (section 4).

2 Classification

PII are heterogeneous and differ from scenario to scenario. Therefore, an automatic learning system has to be developed to detect e-mails containing PII. The classification of such e-mails is a binary text classification problem meaning that there is one class with PII and one without PII. The issue is equivalent to the spam filter problem where e-mails need to be classified in good (ham) or bad ones (spam).

A frequently used algorithm for binary text classification is the Naive Bayes classifier. *Naive* refers to the assumption that the occurrence of the features, in this case the words of the e-mail, are distributed independently. But this assumption does not hold for textual data. The Naive Bayes classifier estimates a probability for each class based on previously classified data. The best class is the one with the highest probability (*maximum a posteriori*) [2]. Thus, for each class a probability is calculated. The prior knowledge $P(c_j)$ of each class is multiplied by the product of all $P(a_i|c_j)$. $P(a_i|c_j)$ is the conditional probability that the term a_i occurs in class c_j based on previously classified data.

$$c_{map} = \arg\max_{c_j \in C} P(c_j) \prod_{i=1}^{n} P(a_i|c_j) \tag{1}$$

Naive Bayes is the basis of most spam filters [3]. The most widely used model of spam filters is based on Graham's and Robinson's ideas [4, 5]. Robinson extended Graham's work by customizing the conditional probability $P(a_i|c_j)$ in order to deal with rare words that occur only in one class. Furthermore, he used the Fisher method to combine the conditional probabilities into a χ^2-distribution with $2n$ degrees of freedom. In his model, an e-mail can be classified as *unsure* when the classifier is not sure about the correct class. The conditional probabilities for the classes c_{ham} and c_{spam} are combined into a χ^2-distribution $\chi^2_{2n} = -2\ln(\prod_{i=1}^{n}(P(a_i|c_{spam})))$.

For the second class a χ^2-distribution can be defined by replacing $P(a_i|c_{ham})$ with $1 - P(a_i|c_{spam})$ because it is $P(a_i|c_{ham}) = 1 - P(a_i|c_{spam})$ due to the binary classification task. The inverse χ^2_{2n}-function χ^{-2}_{2n} is applied to both χ^2-distributions to calculate a score for each class (H and S). The Equation for score H is $H = \chi^{-2}_{2n}(-2\ln(\prod_{i=1}^{n}(P(a_i|c_{spam}))))$.

Finally the two scores are combined into a single score $I = \frac{1+H-S}{2}$ with values between 0 and 1.

We use this approach and substitute the two classes *ham* and *spam* with *PII* and *noPII*. The classifier learns from e-mails marked as *PII* or *noPII* and the calculation of the scores I remains the same. The classification of the e-mails then is done by defining two thresholds t_{noPII} and t_{PII}:

$$class = \begin{cases} \text{PII} & \text{if } t_{PII} > I > 0 \\ \text{unsure} & \text{if } t_{noPII} \geq I \geq t_{PII} \\ \text{noPII} & \text{if } 1 > I > t_{noPII} \end{cases} \quad (2)$$

If the classification returns the class *unsure.* then the user has to do the classification manually. If $t_{noPII} + t_{PII} = 1$, then there is no third class *unsure.*

3 Architecture

The scenario of privacy policies requires to combine the classification of textual information proposed in the foregoing section with usage control and provenance tracking technologies. Hence, we have developed an overall architecture described in the first part of this section forming the framework for these components. Following, the structure of the PII-detector is clarified. An instantiation shows the practical applicability of our approach.

The Usage Control and Provenance Architecture. Our PII-detector is embedded in an architecture to enable usage control and provenance tracking of the sensitive data contained in detected e-mails. A basic usage control architecture consists of a policy enforcement point (PEP), a policy decision point (PDP) and a policy information point (PIP). The PEP is integrated in each application or system where policies have to be enforced in. Every time an event (like "copy" or "delete") is detected, the PEP informs the PDP and asks if the event is allowed or not. The PDP decides this request based on the deployed policies. In the case of data-centered policies [6], the PDP asks the PIP if the container (e.g., files, e-mails) mentioned in the event contains data referred to in one of the deployed policies. If the PDP allows the event to happen, the information flow model of the PIP is updated. Provenance tracking takes advantage of the information flow state represented by the PIP and provides comprehensive information to the person concerned (data subject) [1].

The process of introducing a PII-containing e-mail to the infrastructure is as follows: First, the policy of the container is deployed at the PDP. Second, representations of the containers holding the newly detected PII are created at the Provenance Storage Point (ProSP). Additional meta information (e.g., sender e-mail address) is provided. Third, the PIP is informed about a new representation, connecting the container and the data ID.

The PII-Detector as Part of a Mail User Agent. The processing of incoming e-mails is handled by the mail user agent (e.g., Mozilla Thunderbird) in which the PII-detector is integrated by taking advantage of the agent's plug-in framework. The PII-detector is responsible for three tasks: *Detect* e-mails

containing PII based on the previous described classification model, *label* the e-mails classified as sensitive, and *inform* the usage control infrastructure. For detection, the message body of a new incoming e-mail is tokenized in its single words. The header of the e-mail is omitted, because the potentially sensitive sender and receiver addresses do not provide any information about the sensitivity of the content itself. Furthermore a rule-based classification on header fields (e.g. messages from sender A contains always PII) could lead to a high percentage of false classified e-mails. Next, an analyzer calculates probabilities for each word and combines them with the Fisher method (see section 2). Based on the calculated scores, the e-mail is marked as an e-mail containing PII or not. Finally, the tokens of the new incoming e-mail are added to the existing training data. E-mails clasified as containing PII are labeled by adding a special tag to the header of the e-mail. This labeling is shown to the user via the graphical interface of the mail user agent. By one click, the user can correct the decision made by the classifier.

To show the benefit of our approach, we *instantiated* the PII-detector as the Thunderbird (TB) extension *Thunderbayes4PII*. A PEP for TB has already been developed [7]. Furthermore, spam filter implementations are available. Our implementation is built on *Thunderbayes++*.[1] We selected it because it integrates SpamBayes,[2] an one-to-one implementation of the methodology of Robinson and Graham. We adapted SpamBayes such as only the body and the subject of a message are analyzed for classification. In addition, we replaced the classes *ham* and *spam* by *PII* and *noPII*. Moreover, spam-specific components were deactivated and the GUI was adapted to the PII-detection task.

4 Evaluation

We evaluate the performance of the previously described PII-detector to detect e-mails with PII with a test data set consisting of e-mails with (class *PII*) and without PII (class *noPII*). Our use case is the e-mail communication between customers and the customer support of a telecommunication company. Unfortunately, there are no publicly available e-mails with PII. Hence, we used public data sets which we transformed to an appropriate test set by adding PII.

We crawled the messages of customers from the *Service Forum* of *Deutsche Telekom AG*,[3] where customers ask questions concerning telecommunication services, bills or orders. Although these messages are public, they represent communication between a customer and customer service. Furthermore, the style of writing in these messages is similar to e-mail communication. However, these messages do not contain any PII to identify the customer. Therefore we added artificially generated PII to create a realistic data set for the class *PII*. Depend-

[1] ThunderBayes++ Google Code Project `https://code.google.com/p/thunderbayes/`.
[2] SpamBayes Project Website `http://spambayes.sourceforge.net/`.
[3] Deutsche Telekom AG Service Forum `http://forum.telekom.de`.

ing on the content of the messages, we added PII such as the full address, a customer/access/invoice/order/phone number or banking account data.

For class *noPII* we selected e-mails from the authors' mailbox, e-mails about telecommunication, internet advertisements and marketing. Moreover, e-mails from the *Service Forum* of the Deutsche Telekom not containing PII were chosen. Likewise, we added e-mails from the ENRON data set, which represents internal communication of a company. Table 1 provides an overview of the selected sources.

Table 1. Data Sources

Class PII	**Class noPII**
Messages from Deutsche Telekom Forum	E-mails from the authors' mailbox E-mails from ENRON data set[4] E-mails about internet marketing[5] E-mails from internet advertisement platform (Zanox[6]and Affilinet[7])

The messages from the *Service Forum* were converted into an e-mail format. The e-mails are written in German language, except the e-mails from the ENRON data set, which were translated from English to German. For the evaluation, we used 500 e-mails belonging to class *PII* and 500 e-mails belonging to class *noPII* whereby the data set consists of 1000 e-mails in total. We used a confusion matrix to create measures to evaluate the performance of the classifier [8]. True Positives (TP) and True Negatives (TN) refer to the number of elements, which are classified correctly while False Positives (FP) are predicted as positives, but are actually negatives. False Negatives (FN) are predicted as negative elements which are actually positives. Unsure Negative (UN) and Unsure Positive (UP) are the e-mails, which are classified as *unsure*. The confusion matrix in table 2 summarizes TP, FP, FN, TN, UN and UP.

Table 2. Confusion Matrix

		Predicted class		
		PII	**noPII**	**unsure**
Actual class	**PII**	TP	FN	UP
	noPII	FP	TN	UN

[4] Enron Data `http://enrondata.org/`.
[5] ServiceReport `http://servicereport.eu`.
[6] Zanox AG, `http://www.zanox.com`.
[7] affilinet GmbH, `http://www.affili.net`.

The True Positive Rate ($TPR = \frac{TP}{TP+FN+UP}$) specifies the fraction of e-mails containing PII, which are classified correctly. True Negative Rate ($TNR = \frac{TN}{TN+FP+UN}$) specifies the fraction of e-mails containing no PII, which are classified correctly. The False Positive Rate ($FPR = \frac{FP+UN}{TN+FP+UN}$) specifies the fraction of e-mails containing no PII, which are classified incorrectly. The False Negative Rate ($FNR = \frac{FN+UP}{TP+FN+UP}$) specifies the fraction of e-mails containing PII, which are classified incorrectly. The Error rate measures the fraction of incorrectly classified e-mails in total.

$$Error = \frac{FP + FN + UN + UP}{TP + FP + TN + TP + UN + UP} \tag{3}$$

Accuracy describes the fraction of the correctly classified e-mails in total.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + TP + UN + UP} \tag{4}$$

We performed a *n-fold* cross-validation to evaluate the performance of our detection system. This means that the data set was split into n subsets. *n-1* of them are used for training the PII-detector, the remaining subset was the test data to evaluate it. This was repeated n times, each time with a different training and test set. The average of the results of all evaluations is the performance of the model. We performed an evaluation for *n=5* and *n=10*. This means, the training data in the *10-fold* cross-validation consisted of 900 e-mails and the test data of 100 e-mails. In the *5-fold* cross-validation the ratio was 800 e-mails for training and 200 e-mails for testing.

Furthermore, the thresholds t_{PII} and t_{noPII} were varied. We started with $t_{PII} = 0.1$ and $t_{noPII} = 0.9$ and increased t_{PII} in each iteration by 0.1 and at the same time decreased t_{noPII} by 0.1 (see figure 1).

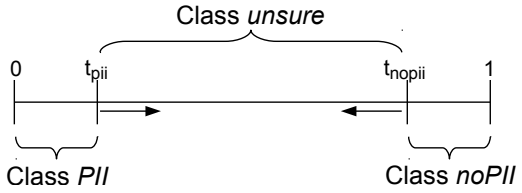

Fig. 1. Variation of t_{PII} and t_{noPII}

Table 3 shows the average results of each iteration. The results of the cross-validation show, that the PII-detector classified e-mails from class *PII* correctly and did not misclassify any e-mails with PII for $t_{PII} \geq 0.4$. The PII-detector misclassified e-mails from the class *noPII* for some values of t_{noPII}. The e-mails from the class *PII* seem to be a homogeneous class and differ strongly from most e-mails of the class *noPII*.

Table 3. Cross-Validation *n=10* and *n=5*

Parameter		**Results n=10**						**Results n=5**					
t_{PII}	t_{noPII}	Err.	Acc.	TNR	FNR	TPR	FPR	Err.	Acc.	TNR	FNR	TPR	FPR
0.1	0.9	0.062	0.938	0.882	0.006	0.994	0.118	0.061	0.939	0.882	0.004	0.996	0.118
0.2	0.8	0.050	0.950	0.902	0.002	0.998	0.098	0.051	0.949	0.898	0.000	1.000	0.102
0.3	0.7	0.048	0.952	0.906	0.002	0.998	0.094	0.048	0.952	0.904	0.000	1.000	0.096
0.4	0.6	0.046	0.954	0.908	0.000	1.000	0.092	0.047	0.953	0.906	0.000	1.000	0.094
0.5	0.5	0.045	0.955	0.910	0.000	1.000	0.090	0.047	0.953	0.906	0.000	1.000	0.094

The results of the evaluation show that our proposed system can detect e-mails with PII very well and has an accuracy of more than 95%. The artificial inclusion of PII in the Telekom data set has not influenced the quality of the evaluation. The classification of the Bayes filter was in all e-mails based on key words in proximity to the PII, but not on the PII itself. Nevertheless, it should be considered that the quality of the PII-detector depends on the quality of the training data. Furthermore, the training and test data consists of e-mails in German language. The results could differ in other languages.

5 Related Work

Text classification is mostly done by learning algorithms. According to the literature, the results of evaluation and comparison of different classification models (Support Vector Machines, k-Nearest-Neighbours, Decision Trees, Naive Bayes, ...) on textual data has shown that Support Vector Machines performed best in the evaluations [9–11]. Still, Naive Bayes classifiers have the advantage to learn new data incrementally.

Access & Usage Control systems need policy specification languages [12]. Usage control can be understood as an extension of access control to the future [13]. Usage control is concerned with how data can be used when it has been accessed to. Usage control systems have been instantiated for several kinds of layers such as Thunderbird [7], Firefox [14], and Windows [15].

Provenance tracking originates in scientific computing community [16]. But in recent work it is also discussed how provenance can improve transparency in the context of privacy [1]. PII is especially relevant in large, unstructured data sets. Hence, a system like the one proposed in this work is a prerequisite to utilize usage control and provenance tracking for the purpose of privacy improvements.

6 Conclusion

We developed a system that is able to detect e-mails with PII. It is embedded into a usage control and provenance architecture. To our knowledge, it is the first approach to detect PII automatically and annotate them with policies. We performed a soundly evaluation for a realistic data set and the results have shown that our PII-detector has an accuracy of over 95%, indicating that the approach

is promising. Up to now, our evaluation is limited to the German language. Further evaluations with data sets in other languages are suggested.

Future work could encompass extending our PII-detector to other data sources like Word documents. In some cases it could be also interesting to have more than two classes, like PII of different degrees of sensitivity.

Acknowledgment: This work was partially funded by Fraunhofer Gesellschaft Internal Programs, Attract 692166.

References

1. Bier, C.: How Usage Control and Provenance Tracking Get Together - A Data Protection Perspective. In: IEEE Security and Privacy Workshops (SPW 2013). (2013) 13–17
2. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval. Cambridge University Press Cambridge (2008)
3. Zdziarski, J.: Ending Spam: Bayesian Content Filtering and the Art of Statistical Language Classification. No Starch Press (2005)
4. Graham, P.: A plan for spam. (2002) `http://www.paulgraham.com/spam.html`, last accessed 15.01.2014,.
5. Robinson, G.: A statistical approach to the spam problem. Linux Journal (107) (March 2003) `http://www.linuxjournal.com/article/6467`, last accessed 15.01.2014.
6. Pretschner, A., Lovat, E., Büchler, M.: Representation-Independent Data Usage Control. In: Proc of the 6th Int Conf, and 4th Int Conf on Data Privacy Management and Autonomous Spontaneus Security (DPM'11). (2011) 122–140
7. Loerscher, M.: Usage Control for a Mail Client (2012) Master thesis.
8. Han, J., Kamber, M., Pei, J.: Data mining: concepts and techniques. Morgan Kaufmann (2012)
9. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. Machine learning: ECML-98 (1998) S.137–142
10. Zhang, T., Oles, F.J.: Text categorization based on regularized linear classification methods. Information Retrieval **4** (2001) S.5–31
11. Aas, K., Eikvil, L.: Text categorisation: A survey. Technical report, Norwegian Computing Center, Oslo (1999)
12. Pretschner, A., Schutz, F., Schaefer, C., Walter, T.: Policy Evolution in Distributed Usage Control. In: Proc of the 4th Int WS on Security and Trust Management (STM 2008), Elsevier 109–123
13. Park, J., Sandhu, R.: The UCON ABC usage control model. ACM Transactions on Information and System Security **7**(1) (February 2004) 128–174
14. Kumari, P., Pretschner, A., Peschla, J., Kuhn, J.M.: Distributed Data Usage Control for Web Applications: A Social Network Implementation. In: Proc of the 1st ACM Conf on Data and application security and privacy (CODASPY), San Antonio, TX (2011) 85–96
15. Wüchner, T., Pretschner, A.: Data Loss Prevention based on data-driven Usage Control. In: Proceedings of the IEEE 23rd International Symposium on Software Reliabilitiy Engineering (ISSRE2012). (2012) 151 – 160
16. Simmhan, Y.L., Plale, B., Gannon, D.: A survey of data provenance in e-science. ACM Sigmod Record **34**(3) (2005) 31–36