Challenges in Classifying Privacy Policies by Machine Learning with Word-based Features

Keishiro Fukushima
Department of Informatics, Kyushu University
Fukuoka, JAPAN
keishiro.fukushima@inf.kyushu-u.ac.jp

Toru Nakamura
KDDI Research
Fujimino, Saitama, JAPAN
tr-nakamura@kddi-research.jp

Daisuke Ikeda
Department of Informatics, Kyushu University
Fukuoka, JAPAN
daisuke@inf.kyushu-u.ac.jp

Shinsaku Kiyomoto
KDDI Research
Fujimino, Saitama, JAPAN
kiyomoto@kddi-research.jp

ABSTRACT

In this paper, we discuss challenges when we try to automatically classify privacy policies using machine learning with words as the features. Since it is difficult for general public to understand privacy policies, it is necessary to support them to do that. To this end, the authors believe that machine learning is one of the promising ways because users can grasp the meaning of policies through outputs by a machine learning algorithm. Our final goal is to develop a system which automatically translates privacy policies into privacy labels [1]. Toward this goal, we classify sentences in privacy policies with category labels, using popular machine learning algorithms, such as a naive Bayes classifier. We choose these algorithms because we could use trained classifiers to evaluate keywords appropriate for privacy labels. Therefore, we adopt words as the features of those algorithms. Experimental results show about 85% accuracy. We think that much higher accuracy is necessary to achieve our final goal. By changing learning settings, we identified one reason of low accuracies such that privacy policies include many sentences which are not direct description of information about categories. It seems that such sentences are redundant but maybe they are essential in case of legal documents in order to prevent misinterpreting. Thus, it is important for machine learning algorithms to handle these redundant sentences appropriately.

CCS CONCEPTS

• **Security and privacy** → **Usability in security and privacy**; • **Computing methodologies** → *Classification and regression trees*; • **Information systems** → Content analysis and feature selection;

KEYWORDS

Privacy Labels, Bag-of-Words Model, TF-IDF, Naive Bayes Classifiers, Support Vector Machines, Random Forests

ACM Reference Format:
Keishiro Fukushima, Toru Nakamura, Daisuke Ikeda, and Shinsaku Kiyomoto. 2018. Challenges in Classifying Privacy Policies by Machine Learning with Word-based Features. In *ICCSP 2018: 2018 the 2nd International Conference on Cryptography, Security and Privacy, March 16–19, 2018, Guiyang, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3199478.3199486

1 INTRODUCTION

As the number of Web services which utilize personal data is increasing, they cause great concern about privacy violation. Therefore, service providers set their privacy policies and obtain consensus from users on how to deal with their private data.

However, it is skeptical that a privacy policy is effective for privacy protection [2] since a privacy policy is difficult to understand in general. To tackle this issue, there exist researches for creating privacy labels, which are used to make policies easier to understand [3]. As such a research, Pontes *et al.* proposed a method, which semi-automatically adds privacy labels to a privacy policy [1]. In this method, pattern matching is used with keywords pre-identified by experts, and thus it is not applicable to privacy policies including new terms.

Instead of a fixed set of keywords, our basic idea is to use machine learning algorithms to identify keywords related to privacy labels. For this purpose, it is natural to use a supervised machine learning algorithm, or equally a classifier. In general, a privacy policy could have many privacy labels. In other words, a policy has many parts, each of which is related to one label. Therefore, we consider classification of sentences in privacy policies, instead of whole policies.

However, it seems to be challenging for such an algorithm to classify privacy policies into some categories. To see this, let us consider the classification of mails into spam or non-spam mails, which is a typical problem of classification. In this case, it is well-known that even a simple algorithm, such as a naive Bayes classifier, can classify mails with high accuracy because there exist some words whose distributions are quite different among spam and non-spam mails. On the other hand, consider to add privacy labels, such as types of information collected, into a privacy policy. In this case, there exist the following challenge: while spam mails are directly written to convey their messages, privacy policies are much more

ICCSP 2018, March 16–18, 2018, Guiyang, China
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6361-7/18/03...$15.00
https://doi.org/10.1145/3199478.3199486

difficult to understand because complex descriptions are necessary to avoid misinterpretation.

In this paper, we compare three popular learning algorithms with different settings for sentence-based classification, and discuss challenges when we develop a system adding privacy labels to privacy policies. There exists three types of learning algorithms: generative models, probabilistic discriminative models, and non-probabilistic discriminative models. We use all of these types. From probabilistic models, we can evaluate each features and thus we can expect that important features are useful when we add privacy labels. Although deep learning algorithms have been achieving very good performance on document classification [4, 5], we do not use them because the amount of data is small for deep learning algorithms.

2 RELATED WORK

Constante *et al.* proposed a method for evaluating the completeness of privacy policies, using NLP and machine learning techniques, where a privacy policy is said to be *complete* if it contains descriptions, which should be explained in privacy policies, such as a description about how to deal with coockies [6]. Guntamukkala *et al.* [7] extends the method in [6] so that a privacy policy is evaluated if it contains descriptions required by some laws, in addition to those defined in [6]. Utilizing crowdsourcing, Terms of Service; Didn't Read (ToS;DR) evaluates privacy policies and offers an add-on for a browser [8]. Zimmeck *et al.* extended their method and mentioned implementation of it as an add-on for a browser [9]. These methods only consider whether a policy contains some descriptions or not, but does not to identify data types or purposes to collect data. Therefore it is difficult for them to be applied to produce or translate to privacy labels. And, a method based on crowdsourcing, such as ToS;DR, heavily depends on human resources and so it is not applicable for a huge amount of privacy policies.

As far as the authors know, the research by Pontes *et al.* is only one which uses NLP and machine learning techniques to identify a purpose to collect data or types of data to be collected in privacy policies. They proposed a method, which semi-automatically translates a privacy policy into privacy labels, using TF-IDF and the Rabin-Karp algorithm, which are a term weighting method in information retrieval and a pattern matching algorithm, respectively. This method, first, calculates TF-IDF values of words in given privacy policies. Then experts choose some words with high TF-IDF values and associate labels with words. Finally, using the Rabin-Karp algorithm, this outputs a label, which is associated with a selected word, if a selected word appears in a privacy policy. If a privacy policy contains one of a selected words, then this method can assign a privacy label. However, this method heavily depends on the selected words by experts and thus it can not assign a label if new words for describing data types or purposes to collect data are used. In addition to that, this method does not make full use of TF-IDF. We will explain this in Seciotn 4.2.

3 METHODS

In this section, we briefly explain learning algorithms and our features based on TF-IDF.

3.1 Learning Algorithms

There exists three types of machine learning algorithms: one is a generative model, such as naive Bayes classifiers, one is a probabilistic discriminative model, such as logistic regression, and the other is non-probabilistic discriminative model, such as support vector machines. We use three algorithms from all types: naive Bayes classifiers, random forests, and support vector machines.

A naive Bayes classifier is a generative model, learning the joint probability of both data and classes. It evaluates $P(C|D)$, the conditional probability that document D's category is category C, given D, using the following Bayes' theorem:

$$P(C|D) = \frac{P(C)P(D|C)}{P(D)} \propto P(C)P(D|C), \tag{1}$$

where we can compute $P(D|C)$ from the training data, in the setting of supervised learning. When we have two categories C_1 and C_2, we compare two probabilities $P(C_1|D)$ and $P(C_2|D)$, and choose the highest one as the category for D.

A random forest is an ensemble learning method, using multiple descition trees, which learn boudaries of data by recursively partitiong the input space. This is not a generative model but a discriminative one. But both a naive Bayes classifier and a random forest are probabilistic approaches.

A suppor vector machine, SVM for short, is a non-probabilistic approach. It is a binary classifier based on a learned boundary, which maximizes the margine between the two classes, that is a discriminative one.

3.2 Features based on TF-IDF

Basically, we use frequencies of words as features. In other words, we assume bag-of-words model. As an extension of this model, we use weighted frequencies, inspired by TF-IDF, which is a standard weighting scheme for indexing terms in the field of information retrieval [10].

Given a set of document and a word, *TF-IDF* is defined as a score of the word against each document, and is the product of the term frequency (TF) and the inverse document frequency (IDF). Roughly speaking, TF is a frequency of the term in a document. Formally, $\mathrm{tf}(t, d)$, which is the value of TF, for a word t in a document d is defined as follows:

$$\mathrm{tf}(t, d) = \frac{n_{t,d}}{\sum_k n_{k,d}}, \tag{2}$$

where $n_{t,d}$ is the frequency of t in d. We normalize the frequency dividing by the sum of all words in d. Roughly speaking, IDF shows that a word t in d appears commonly or rarely in all documents. Formally, $\mathrm{idf}(t)$ for a word t is defined as follows:

$$\mathrm{idf}(t) = \log \frac{N + 1}{df(t) + 1} + 1, \tag{3}$$

where N is the total number of documents and $\mathrm{df}(t)$ is the number of documents containing t. When some word t does not appear in the training data, we have $df(t) = 0$ which causes the zero-division problem. To avoid this problem, we use the smoothing by adding 1 to the frequency of each word [10].

In total, a TF-IDF score of a word against a document represents that the expressive power of the word against the document. If a word is general, such as "this" or "make", $\mathrm{tf}(t, d)$ could be high in d, but $\mathrm{df}(t)$ might be small because such a word prevails in most documents. If a word is specific to a document, then its TF-IDF score must be high. But the score might be small when a word is too specific since its TF value might be small. Therefore, we can expect that TF-IDF scores convey information about word distributions over documents.

Our main idea is that we can use such information. To do so, we have to consider the following two issues when we use supervised learning algorithms: one is how to constract a document and the other is how to modify $\mathrm{tf}(t, d)$ for discriminative models.

For the first issue, it is natural that we create a document from sentences in one category. Then we can expect that distributions of words over categories would improve learning accuracy. However, there exists only serveral categories, that is, documents, in the case of privacy labels. That number is too small, compared to the number of documents in the setting of information retrieval, and thus we can not expect diffrent values of IDF over categories.

To address this issue, we compare the following three settings of how to make a document: to treat a sentence as a document; to create $m \times 32$, where m and 32 denote the number of categories and privacy policies we use, respectively; and to creat three documents by concatenating sentences with the same category label.

For the second one, we can not use $\mathrm{tf}(t, d)$ for SVMs and random forests when we classify unknown data using the trained classifiers if we use a category to create a document because TF value is defined for a document, that is, a category. In other words, TF values contain correct answers about categories. For this issue, we use only training data when we calculate TF-IDF values.

4 RESULTS

In this section, we show experimental results of classification after brief explanation of experimental environments.

4.1 Software, Data and Categories

We have developed naive Bayes classifiers in Python 2.7.11, using janome[1], which is a Japanese language morphological analysis library.

We collected 32 privacy policies written in Japanese and used the following 6 categories: “Item”: data to be collected, “Purpose”: the purpose of a service to collect data, “Who”: entity who collects data, “Del”: data removal, “When”: when data is collected, and “Others”.

We manually add a category label to each sentence. For example, we have added Item to “We collect payment information for clearance process, such as an address or email address”, Purpose to “The data is used for the sake of making statistical data of usage of our services”, and Who to “we may provide encrypted email addresses for our testers to some subcontracting research firm”.

After manually putting labels of categories, we have obtained 383 sentences in Item, 420 in Purpose, 148 in Who, 35 in Del, 38 in When, and 2014 in Others.

We evaluate classification with the F-measure and the accuracy. The F-measure F is the harmonic mean of precision P and recall R:

$$F = \frac{2 \times R \times P}{R + P}, \tag{4}$$

where precision (resp. recall) is the ratio correctly retrieved documents to retrieved documents (resp. relevant documents). F-measure is calculated for each category. For example, consider that we have 150 sentences labeled with Purpose. If a classifier successfully predicts 100 sentences as Purpose among 300 sentences output as Purpose, then we have $P = 100/300$ and $R = 100/150$. The total accuracy T represents how many sentences are successfully classified. For example, consider that we have 500 sentences. If a classifier successfully classifies 400 sentences, then T is calculated as $T = 400/500$.

To calculate these values, we use 5-cross validation.

4.2 Follow-up Experiments of Pontes *et al.*

As described in Section 2, the method proposed by Pontes *et al.* utilizes TF-IDF to identify keywords related to privacy labels. During this process, they treated each privacy policy as a document. In this case, words with high TF-IDF values are specific words to each privacy policies, instead of specific to categories.

The following are the top 15 bi-grams[2] with high TF-IDF values: “target of survey”, “marketing research”, “disclosure and etc”, “target of disclosure”, “target individuals”, “at someone’s request”, “of research”, “the third party company”, “application”, “tester for”, “equity shareholder”, “site for”, “job applicant”, “is disclosure”, and “copy of”. It seems that they frequently appear in privacy policies and therefore might be useful when we want to find privacy policies. However, they are not useful to detect categories[3], such as the purpose to collect data.

4.3 Classification with Naive Bayes Classifiers

In this section, we show several experimental results with naive Bayes classifiers, varying settings for classification.

4.3.1 TF-IDF Values for Feature Vectors. First, we compare two methods for computing feature vectors, where we use word n-grams as features: one is the empirical probability, based on frequency of n-grams, and the other TF-IDF values of them.

We show, for example, the top 10 bi-grams (2-gram) with high TF-IDF values in Purpose category: “in order to”, “in order for”, “of service”, “this service”, “is providing”, “for the sake of”, “information”, “about service”, “provision of” and “service”. We find many phrases generally used to show the purpose to collect data, such as “in order to”. We also find “the purpose of” and “use to”, which express the purpose to collect data in general. Then we can expect that TF-IDF values are good for classification.

Table 1 shows F-measures for each categories and Table 2 total accuracies, where as n of word n-grams, we use 1, 2, and 3. From Table 1, F-measures heavily depend on the data size, where we have 383 sentences in Item, 420 in Purpose, 148 in Who, 35 in Del, 38 in When, and 2014 in Others.

From these two tables, we find that both original and TF-IDF naive Bayes classifiers achieve the worst results in case of 3-grams, and that there is no significant difference between results of 1-gram and 2-gram.

[1] http://mocobeta.github.io/janome/en/

[2] Here these words are translated from Japanese bi-grams, that is, 2-gram, and so some of them are not bi-grams in English.

[3] Of course, we can find good words to detect categories when we exhaustively search words even with low TF-IDF values. But, it is not an automatic process.

Table 1: F-measures for each categories

	Original Naive Bayes Classifiers			TF-IDF Naive Bayes Classifiers		
	1-gram	2-gram	3-gram	1-gram	2-gram	3-gram
Item	0.596	0.561	0.472	0.663	0.641	0.580
Purpose	0.726	0.759	0.669	0.749	0.806	0.729
Who	0.279	0.359	0.429	0.460	0.523	0.494
Del	0.000	0.103	0.154	0.294	0.410	0.320
When	0.000	0.000	0.000	0.085	0.077	0.176
Others	0.860	0.853	0.763	0.866	0.867	0.849

Table 2: Total Accuracies, where "Original NB" (resp. "TF-IDF NB") means original (resp. TF-IDF) naive Bayes classifiers.

	Original NB	TF-IDF NB
1-gram	0.784	0.775
2-gram	0.791	0.798
3-gram	0.667	0.766

As opposed to our expectation, from these tables, we can not conclude that there is significant improvements on naive Bayes classifiers when we use TF-IDF values. One of the reason for this is that the number of documents, which is equal to the number of categories in this case, is not enough large so we could not make full use of TF-IDF values. Another reason might be that the original naive Bayes classifiers contain the basic idea of TF-IDF.

4.3.2 Classification with Fewer Categories. To find difficulties when we classify sentences in privacy policies, now we focus on classification with fewer categories. First we use 3 categories, that is, Item, Purpose, and Others, where sentences with Who, Del, or When labels are included in Others since these three categories contain enough sentences.

We show F-measures in Table 3 and total accuracies in Table 4.

Table 3: F-measures for each categories

	Original Naive Bayes Classifiers			TF-IDF Naive Bayes Classifiers		
	1-gram	2-gram	3-gram	1-gram	2-gram	3-gram
Item	0.653	0.628	0.558	0.603	0.543	0.423
Purpose	0.749	0.745	0.719	0.729	0.717	0.603
Others	0.909	0.911	0.903	0.904	0.873	0.762

Compared to the experiments with 6 categories, evaluation values are improved but the improvement is quite marginal.

We think this is because of the existence of Others, that is, too many sentences in Others could decrease evaluation values. Therefore, we remove Others so that there exist two categories Item and Purpose. In this case, a naive Bayes classifier achieves 0.910 of F-measure of Item, 0.917 of F-measure of Purpose, and 0.914 of the total accuracy, where 1-grams are used as features. These values are quite good, compared to the other results.

Table 4: Total Accuracies, where "Original NB" (resp. "TF-IDF NB") means original (resp. TF-IDF) naive Bayes classifiers.

	Original NB	TF-IDF NB
1-gram	0.846	0.855
2-gram	0.805	0.857
3-gram	0.671	0.839

4.3.3 Classification with SVMs and Random Forests. In this section, we show experimental results of the other two algorithms, where the RBF kernel is used for the kernel function of SVMs and the grid-search is used to decide two parameters γ and C. In the experiments in this section, we only use 1-grams and remove 1-grams whose frequency is quite small.

We show F-measures of SVMs in Table 5 and those of random forests in Table 6. We also show total accuracies of all the three algorithms in Table 7. Among these three algorithms, SVMs using TF-IDF values achive the highest accuracy. For each algorithm, the accuracy of TF-IDF one is higher than the original one. However, this improvement is quite marginal.

5 CONCLUSION

In this paper, we have conducted classification of sentences in privacy policies, using three popular classifiers, where we have adopted the Bag-of-Words model. Generally, classification is not sufficient to achieve automatic generation privacy labels from privacy policies. We think that this is because many parts of a policy are labeled with Others, that is imbalanced data, although they do not directly describe information related to privacy labels. The authors believe that these sentences are direct cause of difficulties for users to understand privacy policies. For imbalanced data, there are popular methods, such as under-sampling. Thus, it is an important feature work to introduce such a method to cope with data in Others.

Table 5: F-measures for each categories with SMVs

	Original SVMs			TF-IDF SVMs		
	recall	precision	F-measure	recall	precision	F-measure
Item	0.666	0.686	0.676	0.624	0.718	0.668
Purpose	0.779	0.820	0.799	0.769	0.848	0.806
Others	0.928	0.915	0.921	0.943	0.907	0.924

Table 6: F-measures for each categories with random forests

	Original RF			TF-IDF RF		
	recall	precision	F-measure	recall	precision	F-measure
Item	0.445	0.758	0.561	0.416	0.863	0.561
Purpose	0.690	0.873	0.771	0.695	0.921	0.792
Others	0.967	0.872	0.917	0.983	0.866	0.921

Table 7: Total accuracies of all algorithms, where "NB" and "RF" represent naive Bayes and random forest, respectively.

Algorithm	total accuracy
Original NB	0.846
TF-IDF NB	0.855
Original SVM	0.874
TF-IDF SVM	0.879
Original RF	0.863
TF-IDF RF	0.872

REFERENCES

[1] Diego Roberto Gonçalves de Pontes and Sergio Donizetti Zorzo. PPMark: An Architecture to Generate Privacy Labels Using TF-IDF Techniques and the Rabin Karp Algorithm. In *Information Technology: New Generations*, pages 1029–1040. Springer, 2016.

[2] Aleecia M. McDonald and Lorrie Faith Cranor. The Cost of Reading Privacy Policies. *Journal of Law and Policy for the Information Society*, 2008.

[3] Patrick Gage Kelley, Joanna Bresee, Lorrie Faith Cranor, and Robert W Reeder. A nutrition label for privacy. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, page 4. ACM, 2009.

[4] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level Convolutional Networks for Text Classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, pages 649–657, 2015.

[5] Yoon Kim. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751, 2014.

[6] Elisa Costante, Yuanhao Sun, Milan Petković, and Jerry den Hartog. A machine learning solution to assess privacy policy completeness(short paper). In *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society*, pages 91–96. ACM, 2012.

[7] Niharika Guntamukkala, Rozita Dara, and Gary Grewal. A Machine-Learning Based Approach for Measuring the Completeness of Online Privacy Policies. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 289–294. IEEE, 2015.

[8] Terms of Service; Didn't Read (ToS;DR). http://tosdr.org/ index.html.

[9] Sebastian Zimmeck and Steven M Bellovin. Privee: An Architecture for Automatically Analyzing Web Privacy Policies. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 1–16, 2014.

[10] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schuetze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.