Automatic Summarization of Privacy Policies using Ensemble Learning

Noriko Tomuro
College of Computing and Digital Media (CDM)
DePaul University
Chicago, IL USA
tomuro@cs.depaul.edu

Steven Lytinen
College of Computing and Digital Media (CDM)
DePaul University
Chicago, IL USA
lytinen@cs.depaul.edu

Kurt Hornsburg
MobileEvolution
Vienna, Austria
kurt.hornburg@mobileevolution.eu

ABSTRACT
When customers purchase a product or sign up for service from a company, they often are required to agree to a Privacy Policy or Terms of Service agreement. Many of these policies are lengthy, and a typical customer agrees to them without reading them carefully if at all. To address this problem, we have developed a prototype automatic text summarization system which is specifically designed for privacy policies. Our system generates a summary of a policy statement by identifying important sentences from the statement, categorizing these sentences by which of 5 "statement categories" the sentence addresses, and displaying to a user a list of the sentences which match each category. Our system incorporates keywords identified by a human domain expert and rules that were obtained by machine learning, and they are combined in an ensemble architecture. We have tested our system on a sample corpus of privacy statements, and preliminary results are promising.

General Terms
Privacy, Languages, Algorithms, Experimentation

Keywords
Privacy Policy; Natural Language Processing; Machine Learning

1. INTRODUCTION
A wide variety of companies require their customers to agree to a Privacy Policy (or Terms of Service) when purchasing products and services. These policies are often quite intrusive, including conditions about how a customer's personal information will be gathered and retained, shared by subsidiaries or sold to other companies, and so on. The vast majority of customers agree to these policies without reading them carefully if at all [1]. This means that for many people, their personal data is stored, used, and/or shared without them being aware that this is the case.

One potential approach to helping a customer understand privacy policies is through text summarization. While automated approaches to text summarization are certainly not infallible, a customer would be more likely to read a short, computer-generated summary of a privacy policy than the full text. We have developed a prototype of an automatic text summarization system which is specific to privacy policies. The system generates a summary by identifying important sentences in a policy, and presenting those sentences to a user according to which of 5 "statement categories" is addressed by each sentence. Although there are only few previous works which attempted automatic privacy policy analysis (e.g. [2][3]), our system is unique in several ways. First, it incorporates both the knowledge of a human domain expert (provided as keywords), and the knowledge obtained automatically through machine learning. The two types of knowledge are combined in an ensemble architecture to exploit the synergy between them. Second, our system represents both types of knowledge in the form of if-then rules, which are human-readable and easy to understand. Also the system is implemented as a web application, and publicly accessible (http://slytinen-ntomuro.rhcloud.com/index.jsp). We conducted a preliminary evaluation of the system's performance. The results were promising – with the initial set of rules which are yet to be refined, the system showed relatively high recall and precision.

CODASPY'16, March 09-11, 2016, New Orleans, LA, USA.
ACM 978-1-4503-3935-3/16/03.
http://dx.doi.org/10.1145/2857705.2857741

2. PRIVACY SUMMARIZER SYSTEM
The main goal of the system is to summarize a privacy policy by extracting key sentences which address its major points, and displaying them in a concise manner. We identified 5 privacy categories as the most important, essential information to understand a privacy policy, and defined our own *Statement Categories* in the form of questions:

- Statement 1 (Clear Purpose): For what purposes does the company use personal information?
- Statement 2 (Third Parties): Does the company share my information with third parties?
- Statement 3 (Limited Collection): Does the company combine my information with data from other sources?
- Statement 4 (Limited Use): Will the company sell, re-package or commercialize my data?
- Statement 5 (Retention): Will the company retain my data? What is their retention policy?

We created our dataset of privacy policies by downloading the privacy policy page of many major companies from a wide range of business areas. So far there are a total of 76 policies in the dataset. A domain expert manually annotated the individual sentences in (randomly selected) 25 policies with the statement categories. This annotated subset corpus consisted of 335 sentences which are statement category 1-5, and 4424 sentences which didn't address any of the categories. We used this corpus to build our system, as described in detail in section 3.

Our system is implemented as a web application. Figure 1 shows its interface. The system incorporates all 76 policies, and they are shown in a dropdown menu; when the user selects a statement, the summary is displayed as shown in Figure 2.

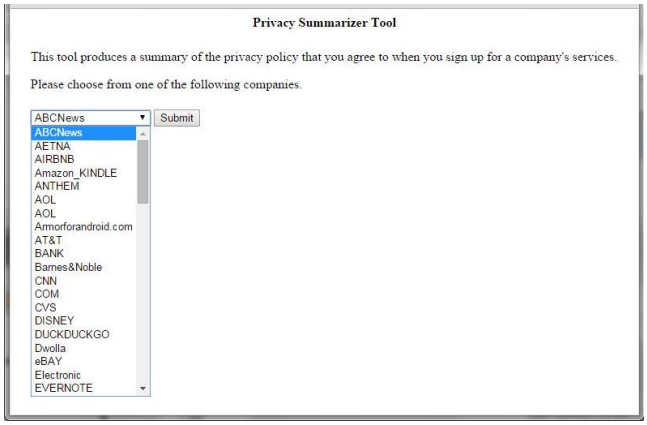

Figure 1. User Interface of the System.

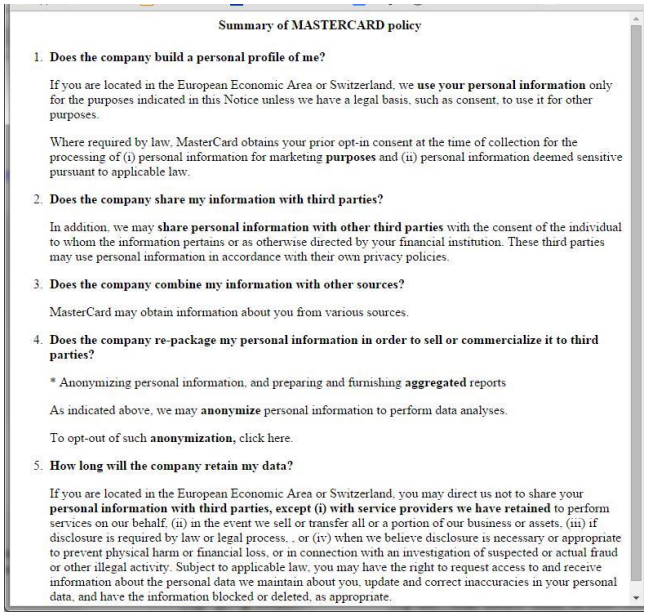

Figure 2. Summary of a policy extracted by the system.

Note that a summary is a list of sentences which the system extracted from the policy files, as is, with no modification (i.e., sentences themselves are not shortened). Also currently the development is at an early stage, and the interface is still rough. Furthermore, at this moment the system displays all sentences that matched each of the 5 statement categories. We plan to improve the interface, and display just one or two *best* sentences that matched the statement.

3. DETAILS OF THE SYSTEM

The system is composed of three components which are organized in a two-level ensemble hierarchy, as shown in Figure 3. The first level consists of two components – a pattern matcher based on manually crafted patterns (the "Keyword Matcher"), and another matcher based on patterns automatically derived by a Machine Learning algorithm (the "JRIP matcher"). Then the component on the second level (the "Combiner Classifier") receives the output of the two level-1 matchers and produces the final result, which is the statement category of the sentence fed in the system (category 1/2/3/4/5 or 0 for no-category).

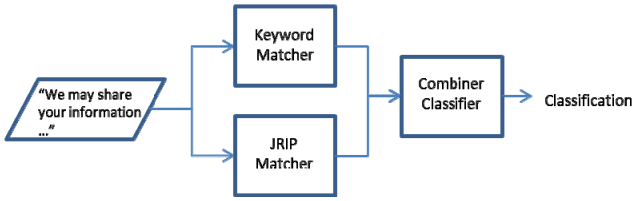

Figure 3. Schematic components of the system

3.1 Keyword Matcher

The Keyword matcher is based on the keywords provided by a human domain expert. The keywords are manually crafted into several 'patterns' (in the form of regular expressions) for each of the 5 statement categories.

As an example, the keywords provided by the human expert for statement 3 were as follows:

KEY WORDS:
Combine, supplement, associate, in conjunction, link
Personal information, your information, the information
Third-parties, partners, other companies, other sources

And the regular expressions written for this category included the following pattern:

(combin|supplemen|associa).{0,100}(third|other|informat|data)

Note that ".{0.100}" (occurrence of any character (.) between 0 and 100 times) is an example effort to specify the allowable distance between two keywords (where words in an enclosed parentheses in a pattern are essentially synonyms). We observed most keywords appear in a sentence at the subject or main verb or direct object position. So we wrote patterns to specify that two keywords should appear close to each other (in a local context) to avoid spurious false-positive matches.

For a given sentence the keyword matcher applies all patterns in the 5 sets of patterns (statement category 1-5) and produces 5 resulting values – one for each category. The value is a yes or no, indicating whether or not any of the patterns of the category matched the sentence. Note that, with this scheme, a sentence could match with more than one category.

3.2 JRIP Matcher

The JRIP matcher is based on the rules derived (automatically) by a Machine Learning (ML) algorithm called RIPPER [4]. We used a particular implementation of the algorithm in a ML tool called Weka [5] (in which the algorithm is named JRIPPER).

RIPPER is a classification algorithm, which classifies a given input instance into one of the predefined target categories. We first trained the algorithm using the subset corpus of 25 annotated policies, and obtained a set of 'rules' for each category. The derived rules are a model, in particular a decision rule expressed in the form of an if-then statement and keyed by the words in the sentence. Ordering of the selected words is unimportant for the JRIPPER-derived rules. For example, the rules for statement 3 included the following.

```
(combine >= 1) and (group <= 0) => category=3
(associate >= 1) and (account <= 0) and (access <= 0)
   and (device <= 0) => category=3
(combine >= 1) and (datum >= 1) => category=3
```

Note the values appearing with a word with a <= or a >= are a number of occurrences of the word in the given sentence, and an expression "category=X" after the symbol => is the decision of the rule – the statement category X. For example, the first rule above indicates that a sentence containing one or more occurrences of the word "combine" and not containing "group" should be categorized as statement category 3.

Note that we trained the JRIPPER algorithm using sentences of the categories 1 through 5 and excluded category-0 sentences. We did so because only 5% of the sentences were identified as relevant (category 1-5), and the vocabulary would have otherwise been dominated by terms from category-0 sentences.

Similar to the Keyword matcher, the JRIP matcher applies all 5 sets of rules (statement category 1-5) to each sentence and produces 5 yes/no resulting values. Also just like the Keyword matcher, a sentence could match with more than one category.

3.3 Combiner Classifier

The combiner classifier on the second level receives the output of the two matchers on the first level (totaling 10 yes/no values) and produces the final output – the statement category of the given sentence which the system predicts (1/2/3/4/5 or 0). To develop a model for this classifier, we used the RIPPER algorithm again because of the interpretability of the generated rules. But for the training data, we included some category-0 sentences so that the classifier would learn to identify irrelevant sentences as well as relevant (category 1-5) ones.

As an example, the generated rules included the following. Note that "regexpN" represents the Keyword matcher's result for the statement category N, while "jripN" represents the JRIP matcher's result for category N. Note also that the final classification is mutually exclusive – only one category for a sentence (and 'category 0' was one of the options).

```
(regexp5 = yes) => category=5
(jrip3 = yes) => category=3
(regexp3 = yes) and (jrip1 = no) and (jrip4 = no) => category=3
(regexp2 = yes) and (regexp1 = no) => category=2
```

4. EVALUATION

To evaluate the system, we used the same dataset used for the Combiner classifier, which included a 5% random sampling of the category-0 sentences in addition to the 335 category 1-5 sentences (thus the total number of sentences was (4424*.05) + 335 = 221 + 335 = 556). The table below shows the results for category 1-5. Note that these were obtained by a 10-fold cross-validation.

Meanings of the table's columns are as follows.

- # Actual – The number of sentences of the category, as labeled by the human domain expert.
- # TP (True-Positive) – The number of sentences which the system correctly classified as this category.
- # FP (False-Positive) -- The number of sentences which the system incorrectly classified as this category.
- Recall -- # TP / # Actual. The ratio of the true sentences recalled by the system.
- Precision -- # TP / (#TP + # FP). The ratio of the true sentences in all sentences classified as this category by the system.

Table 1. Results of cross-validation test

Statement	# Actual	# TP	# FP	Recall	Precision
1	81	49	16	0.605	0.754
2	87	30	9	0.345	0.769
3	57	42	13	0.737	0.764
4	59	38	14	0.644	0.731
5	51	41	13	0.804	0.759
Total	335	200	65	0.597	0.755

Based on the results, we can see that approximately 6 in 10 relevant sentences (as judged by the human expert) are correctly labeled by the system. Statement 2 seems very difficult to identify, with recall = 0.345. Overall, precision is somewhat higher than recall; about 3 in 4 sentences (on average) marked as relevant by the system are correctly classified.

5. CONCLUSIONS AND FUTURE WORK

We plan to continue working and improve the system in the future work. The most immediate issue is to reduce false-positives – when irrelevant/category-0 sentences are identified as relevant (category 1-5). The regular expressions which we have crafted are particularly overproductive, and we expect to be able to achieve higher precision with further refinement of these rules. We also plan to develop a component which utilizes non-lexical features (e.g. sentence length, position of a sentence relative to document headers) and to add this as a third module in the first level of our system, to be incorporated into the learning ensemble.

6. REFERENCES

[1] Jensen, C., and Potts, C. 2004. Privacy Policies as Decision-Making Tools: An Evaluation of Online Privacy Notices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.*

[2] Constante, E., Sun, Y., Petkovic, M. and den Hartog, J. 2012. A Machine Learning Solution to Assess Privacy Policy Completeness. In *Proceedings of the ACM Workshop on Privacy in the Electronic Society*, 91-96.

[3] Zimmeck, S. and Bellovin, S. 2014. Privee: An Architecture for Automatically Analyzing Web Privacy Policies. In *Proceedings of the 23rd USENIX Security Symposium*.

[4] Cohen, W. 1995. Fast Effective Rule Induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, 115-123.

[5] Witten, I., Frank, E. and Hall, M.. 2011. *Data Mining: Practical Machine Learning Tools and Techniques*, Third Edition. Morgan Kaufmann.