PEEP - An Information Extraction based approach for Privacy Protection in Email

Narjes Boufaden, William Elazmeh, Yimin Ma, Nour El-Kadri, Stan Matwin, Nathalie Japckowicz
School of Information Technology and Engineering
University of Ottawa
Ottawa, Canada K1N 6N5

Abstract

This paper presents a privacy compliance engine that monitors emails generated in an organization for violation of privacy policy of this organization. Our architecture includes four components: a domain knowledge defining the entities and private information we are dealing with, a pre-analysis component that extracts header information and segments the emails, an information extraction system that extracts private information, and an inference engine that matches the information extracted against a set of compliance rules. The compliance engine warns the sender about possible violations. A prototype has been developed for a university setting. Early empirical tests produced an F-score of 69.3%.

1 Introduction

Data privacy is one of the major societal concerns surrounding Information Technology. Most countries have introduced, in the last several years, data privacy laws. The HIPAA act in the US, its analog Bill 31 in Ontario, and the PIPEDA (Privacy Information Protection in Electronic Documents Act) law in Canada are examples of such legislations. Legal remedies, however, only intervene after privacy of individuals has been breached, and never prevent it. Only technical solutions can stop disclosures of private data when it happens. As email is the main tool of many intra- and inter-organizational communications, it also becomes an instrument of privacy violations. These violations may often be a result of human error: a mistaken alias address in the 'cc' field of a message may disclose private information to thousands of unauthorized recipients.

Although this is a significant problem, solutions proposed in this field do not seem to go beyond the lexical level for detecting and matching data against encoded privacy rules. For instance, Vericept\(^1\) detects the presence of social security numbers, credit card numbers, and other specific identifiers in messages, yet it is clear that detection of privacy violations often requires inference. Privacy rules must be connected with the knowledge about the people and the types of information involved. It is therefore tempting to introduce knowledge-based representations and information extraction (IE) techniques into privacy compliance systems.

Our work is part of an ongoing Privacy Enforcement in Email Project (PEEP)\(^2\) that aims to develop a privacy compliance system, monitoring email in a large organization (e.g. a health care provider, or a university) for potential privacy breaches. In this paper we address the privacy violation in an academic setting where private information is student identification numbers (ID), names and grades for a particular course. We describe our approach, the important design decisions that we have made developing the PEEP architecture, and the early empirical evaluation of our solution.

In particular, we focus on the use of ontologies to describe domain knowledge and constraints and the implementation of an IE engine for emails. The ontology gives a formal description of the bits of information that might be involved in an information breach, whereas, the IE system gives the contextual setting of those information by identifying “to whom the information belongs”. Hence, given some privacy rules, the email recipient’s identity and the private information extracted from the message, it becomes possible to check if there was any privacy violation.

Next section describes the data used in this project. The four component-system architecture is presented

\(^1\)http://www.vericept.com

\(^2\)http://www.peep-project.org
The following student should get 80 on A4, could you please change it?

FirstName1 LastName1 (xxxxxxx) 80

Thanks

Sender

Percentage | Emails about marks
---|---
Repetitions | 0.03%
Misspellings | 1.13%
Ungrammatical utterances | 10.5%

Table 1: Percentage of repeated words, misspellings and ungrammatical utterances found in 93 emails in the topic of “Assignment marks”.

Some emails contain multiple replies which were sorted chronologically to keep the overall context of all the information exchanged for IE purpose.

Ungrammatical utterances are those where either the subject, verb or object is missing or misplaced. Figure 1 shows an example of ungrammatically, where verbs are missing in both utterances of the email body FirstName1 LastName1 (xxxxxxx) 80.

3 Architecture of the system

The privacy compliance engine is composed of four components. The first one is the domain knowledge with a domain ontology describing basic concepts involved in the privacy checking process, an information access ontology defining types of access privileges and a database containing organization information. The second one is a pre-analysis module that extracts sender/recipient information and does the segmentation of the email body for IE purpose. The third component is the IE system which extracts private information from the email body. The fourth component detects privacy breaches. It uses the information extracted from the email body, the recipient/sender information extended with additional information from the database, and a set of privacy rules linking concepts from the domain ontology and information access ontology. Figure 2 shows the four component architecture of the privacy compliance engine.

4 Preprocessing

This stage is divided into three parts:

1. the first part extracts sender/recipient information from the email header. It provides a list of predicates in the following format:

<table>
<thead>
<tr>
<th>Percentage</th>
<th>Emails about marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repetitions</td>
<td>0.03%</td>
</tr>
<tr>
<td>Misspellings</td>
<td>1.13%</td>
</tr>
<tr>
<td>Ungrammatical utterances</td>
<td>10.5%</td>
</tr>
</tbody>
</table>

in section 3. The pre-processing stage, is detailed in section 4. The domain knowledge, is described in section 5. Section 6 describes the IE system and the privacy checking engine is presented in section 7. Finally, we state conclusions and future work in section 8.

2 The Data

Email texts fall into unstructured text category which are neither rigidly formatted nor always composed of grammatical sentences [10]. In some aspects, they are similar to manually transcribed spontaneous speech. They don’t always have explicit punctuation, and they contain some disfluencies such as repetitions of words and omissions\(^2\). They also have misspellings and acronyms which need to be translated to the appropriate words. An example of these email texts is given in Figure 1.

Our corpus is composed of 94 emails talking about assignment marks. They are mostly exchanges between students and professors or between professor and teaching assistants. Some talk about student marks in general, whereas other are personal and contain personal information. To evaluate the overall privacy compliance engine, we changed the type of some recipients to deny access privileges to student private information such as mark.

\(^2\)Omissions occur when a word is missing in an utterance.
Figure 2: Architecture of the privacy compliance engine.

sender(person(FirstName, LastName),
  email(emailAddress)).
recipient(NumberOfRecipient,
  person(FirstName, LastName),
  email(emailAddress)).

where the predicate `person` gives the sender or recipient first name and last name whereas the predicate `email` gives the person’s email.

2. The second part deals with abbreviations. It translates abbreviations such as A4 to fourth assignment or TA to teaching assistant.

3. The third part addresses the segmentation of email bodies by attempting to assemble the verb and its arguments in the same line. This step is an important one because standard IE approaches rely on the syntactic relation “subject-verb-object” to extract relevant information and identify the roles of each argument (subject and object). The segmentation was done using two features:
   - any type of punctuation including consecutive dots and question marks.

5 Domain knowledge

5.1 The database

To represent domain knowledge about entities mentioned in emails, we build a dedicated database. Seven tables and four relations, such as “student-registered-course”, were defined. An example of the tables defined and some of their attributes are shown in table 2.

The tables were implemented in Prolog to simplify the database accesses. However, in future work we plan to develop a database with mySQL in addition to an interface between the Prolog privacy checking engine and the MySQL database.

5.2 Domain ontology

The domain ontology is a hierarchical organization of the main objects described in the database. There is a direct mapping between the concepts of the domain ontology and the database objects such as student and staff (2). However, the database gives additional information related to relations between tables, e.g. “student-registered-course”. We use a domain ontology to organize table objects to assist the semantic tagging and learning extraction patterns [3].

The ontology is a three layer tree with two main classes. The physical entity class gathers every physical entity forms the database such as person, student and staff, whereas the conceptual entity class is a general class under which are courses and programs. Figure 3 shows a part of the is-a domain ontology.

5.3 Information access privilege ontology

The information access privileges ontology is a hierarchical organization of the attribute tables, where the organization is based on types of access privileges de-
pending on the relation “who has the right to access what”. Types of persons are drawn from the domain ontology and access rights are defined by the the Council of Ontario Universities guidelines on freedom of information and privacy protection 3 and the University of Guelph guidelines 4. The Figure 4 shows a part of the information access privilege ontology.

Each access class specifies a list of attributes to be accessed by any individual of the respective class, such as staff versus Staff Access. Axioms (privacy rules) are used to link domain ontology person subclasses and access classes.

6 Information Extraction system

Information extraction is about finding and structuring relevant information in a text given a particular domain. In the academic context, relevant information is, for example, student IDs, names, addresses, and assignment marks.

We developed a three-stage IE system that starts with a shallow parsing of the email body to detect noun groups, numbers and verbs. The second stage is the semantic tagging which relies on the domain ontology. The third stage extracts individual facts by first learning the extraction patterns and then match the patterns against the semantically tagged email. The output of the system is a set of relations and facts in Prolog format.

6.1 Shallow parsing

The shallow parsing was done with the CASS partial parser of Steven Abney [1] and the part-of-speech tagging with the Brill transformational tagger [4]. Candidates to be tagged are noun groups np and verbs vp. Because of the ungrammaticality encountered in emails, many errors occurred when parsing large constructions. So, we reduced the set of grammatical rules used by CASS to cover only minimal chunks and discard large constructions such as VP → H=VX O=NP? ADV* or noun phrases NP → NP CONJ NP.

6.2 Semantic tagging

This task goes beyond named entity extraction (NEE) [6]. It annotates words that are instances of the ontology such as the named entities persons and numbers, but also verbs such as score, receive and expressions such as assignment, mark to characterize the context of relevant information which are in this case attributes of the relation “the assignment mark X of student Y”.

6.2.1 Approach

This process takes every chunk provided by the parser and looks for the first match with a concept instance. The match is based on the word and its part-of-speech. When a match succeeds, the semantic tag assigned is the concept of the instance matched. Then, the semantic tag of the head is propagated to the whole chunk as shown in Figure ??.

Matching step 1: . . . SN: fourth assignment . . .

Propagation step: . . . SN: fourth assignment . . .

Figure 5: Output of the semantic tagging. The semantic tag of the head “assignment” is propagated to the whole chunk.

6.2.2 Experiment and results

The semantic tagger was tested on 3978 words and expressions and the precision and recall scores are given in table ??.

The Fscore of the semantic tagger is comparable to those reported in the proceeding of the seventh Message Understanding Conference (MUC7) [5] for the NE
Table 3: Recall (Rec.), precision (Prec.) and Fscore of the semantic tagger.

<table>
<thead>
<tr>
<th>Words</th>
<th>Rec.</th>
<th>Prec.</th>
<th>Fscore</th>
</tr>
</thead>
<tbody>
<tr>
<td>3978</td>
<td>95.0%</td>
<td>94.4%</td>
<td>95.5%</td>
</tr>
</tbody>
</table>

However the major source of errors occurs with numbers\(^5\) because of the different formats they came in (for example 30/40 as opposed to 30-40 or 30).

6.3 **Extraction of individual facts**

This stage is divided into two parts. The first part has to do with learning extraction patterns. It uses Markov models to learn relevant sequences of semantic tags along with their thematic role. This stage allow the detection of the target relation “the assignment mark X of student Y”, while the second part is the extraction of individual facts, X and Y, using the extraction pattern learned.

Our approach is motivated by three reasons:

- The size of our corpus which is too small to allow learning at the lexical level. So, we used word classes which are concepts from the domain ontology.
- The ungrammatical sentences which make syntactic parsing too inaccurate and incomplete to infer thematic roles.
- Markov models are an efficient way to model observation sequences of various length. It is an easy way to introduce wild-card states and empty states that can handle repetition of words and omissions [7].

6.3.1 **Experiment and results**

We annotated 85 sequences of semantic tags generated by the semantic tagger with the thematic roles. We trained a first order and second Markov models and evaluated the learning process using the “leaving one out” cross validation method [9]. Table 4 shows the average of the recall, precision and Fscore for each model.

<table>
<thead>
<tr>
<th>Model</th>
<th>Rec.</th>
<th>Prec.</th>
<th>Fscore</th>
</tr>
</thead>
<tbody>
<tr>
<td>First order</td>
<td>0.68</td>
<td>0.64</td>
<td>0.67</td>
</tr>
<tr>
<td>Second order</td>
<td>0.73</td>
<td>0.71</td>
<td>0.72</td>
</tr>
</tbody>
</table>

Table 4: Recall (Rec.), precision (Prec.) and Fscore of the first and second order Markov models.

Our approach is motivated by three reasons:

\(^5\)The major source of errors reported in the MUC proceedings are proper names.

The second order model shows better results than the first order model. The analysis of the output shows that classification errors occur when there was no informative context, for example missing keywords such as the verbs receive and get or nouns such as mark or assignment. Most common error was with numbers being identified either as course codes whereas they were referring to marks or student IDs and marks being referred as course codes or student IDs.

In both cases including a pre-tagging stage before information extraction would help to reduce those errors, for example by checking the number of digits to compare student IDs with course codes.

6.3.2 **Extraction of individual facts**

We evaluate the IE system by choosing the Markov model with the closest Fscore to the average Fscore given in table 4 and integrated the model in the privacy compliance engine. Since the overall privacy compliance system was developed in Prolog, we translated each fact and relation into Prolog predicates as shown in figure 1.

The evaluation was made for the 94 emails and the results are shown in table 5.

<table>
<thead>
<tr>
<th>Semantic tagging</th>
<th>Relations</th>
<th>Rec.</th>
<th>Prec.</th>
<th>Fscore</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual</td>
<td>153</td>
<td>37.9%</td>
<td>51.3%</td>
<td>46.3%</td>
</tr>
<tr>
<td>Automatic</td>
<td>85</td>
<td>68.2%</td>
<td>51.3%</td>
<td>58.5%</td>
</tr>
</tbody>
</table>

Table 5: Recall (Rec.), precision (Prec.) and Fscore of the IE system on 94 emails.

For the evaluation we considered relations “the assignment mark X of student Y” involving a pronoun (he, I or you) to be correct as long as the pronoun refers to the right person.

The first evaluation was done on 153 relations extracted from the email bodies by a human annotator, whereas the second evaluation was done on 85 relations inferred by a human annotator from the semantic tagger output. On the one hand, it is clear that semantic tagger misses had dramatic consequences on the IE performance. In particular, since most of the semantic tagging errors occurred with numbers, many “the assignment mark X of student Y” relations could not be detected. On the other hand, the results based on the semantic tagger output are consistent with those of the learning stage, since errors generated in the learning process would occur on the IE process.
7 Privacy checking engine

The last component of the privacy compliance engine is the privacy checking engine. It takes as input the relations extracted and a set of facts provided by the preprocessing component. The engine matches the set of facts and relations against a list of privacy predicates and outputs a violation flag when there is a potential information breach. Privacy rules are prolog predicates that link a particular domain ontology class to a particular type of access information and are designed to prove a valid access granted to particular database attributes.

7.1 Approach

The privacy checking engine is a three stage process:

1. The first stage takes the sender/recipient information and extends them with additional information from the database. In particular, the type of the sender and recipients such as a teacher or an administrator are extracted from the database.

2. The second stage uses the information generated by the first stage to check the access right of the sender/recipients.

3. The last stage matches those information along with the data extracted from emails and generates the violation flag when it applies.

Figure 1 shows the different stages of the privacy checking process on an email talking about upgrading a mark. This email was intended the teaching assistant of this course. However the recipient listed in the email was the wrong one. Therefore, the system fails to identify the domain ontology class of the recipient and defaults to the most general class person. In the rule shown above, the left argument (person) is the domain ontology class of the recipient. He is granted a public access privilege to information. However, the body of the email identifies the students’ names and grades, therefore triggering a “Grade Access Violation”.

While in our previous work [2] the assumption of correct input was true (the information was extracted manually from the emails and was limited to person names, addresses and emails). In these experiments it is no longer true. For example, the figure 6 shows a noisy output where the predicate mark-student(person([‘student’]),mark(‘80’)) has been extracted despite the fact that student is a common noun.

To deal with the noise introduced by the IE system, we rely on filters that check predicate arguments and
<table>
<thead>
<tr>
<th>Email class</th>
<th>Number</th>
<th>Recall</th>
<th>Precision</th>
<th>Fscore</th>
</tr>
</thead>
<tbody>
<tr>
<td>Violation</td>
<td>15</td>
<td>20%</td>
<td>20%</td>
<td>20%</td>
</tr>
<tr>
<td>No violation</td>
<td>79</td>
<td>72.1%</td>
<td>89.1%</td>
<td>79.8%</td>
</tr>
<tr>
<td>Average</td>
<td>94</td>
<td>63.8%</td>
<td>75.9%</td>
<td>69.3%</td>
</tr>
</tbody>
</table>

Table 6: Recall (Rec.), precision (Prec.) and Fscore of the privacy checking engine

verify inconsistency such as course codes being student IDs or names not listed in the student table. Besides, we build rules that attempt to satisfy the least conditions required to infer an access privilege.

7.2 Experiments and results

We experiment with 94 emails that had been processed by the IE system. Fifteen of them are considered violation of privacy. Table 6 shows the precision, recall, and F-score of the privacy checking engine.

The checking engine produces poor results for those emails consedered as violations. An analysis of the inference process shows that most errors occured when the IE extracts incomplete relations. Hence private information can’t be attached to a particular person and in this case is considered as public.

8 Conclusion and Future work

In this paper we addressed the privacy checking problem using IE techniques and domain knowledge that captures bits of information involved in different privacy violations scenarios and constraints that define those violations.

Using an ontology to model domain knowledge and constraints is consistent with the EPAL approach, which makes an ontology a prerequisite for the representation of privacy rules.

We addressed the pattern learning in a different way from related works [8, 10]. For example, Soderland used semantic classes to learn regular expressions from on-line rental adds [10]. His system extracted individual facts with an Fscore around 94%. However, the adds are shorter texts with a more restricted format than our emails. The closest work to ours was developed for the CALO project [7], which aims to extract information about people and other entities appearing in email streams. Yet, to our best knowledge the results of the IE system were not published.

Even though the results achieved by the overall privacy compliance engine are encouraging, there is room for improvement, in particular in the pre-processing part and the semantic tagging stage.

In future work, we plan to tackle two issues. The first on is the integration of the EPAL language in our design by translating the information access privileges ontology into an EPAL description, so it would be expressed in a standard way, allowing its use for other privacy applications. The second issue is applying our system to the health care domain. We are collaborating with The Ottawa Hospital (TOH) on this application of research.

References
