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

**Narjès Boufaden***, **William Elazmeh***, **Stan Matwin*+**, **Yimin Ma***, **Nour El-Kadri***, **Nathalie Japkowicz***

*School of Information Technology and Engineering
University of Ottawa, Ottawa, Canada K1N 6N5.
+ Institute for Computer Science,
Polish Academy of Sciences, Warsaw, Poland.

## Abstract

This paper presents a privacy compliance engine that monitors outgoing emails in an organization for violation of privacy policy of this organization. Our architecture includes four components. Domain knowledge defines roles in the organization and disclosure privileges associated with types of private information we are dealing with. A pre-analysis component extracts sender and recipients identity from the header and segments the emails. Then an information extraction system extracts private information, and an inference engine 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 a major societal concerns surrounding Information Technology. Many countries have introduced data privacy laws and legislations. 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, roles[2] in an organization, and the types of information involved, table attributes in a database. 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 outgoing emails 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 propose a role based approach similar in some aspects to the Role Based Access Control model (RBAC) [7] which associates access rights with roles in an organization. In our approach we use an ontology to model roles in the academic domain and attach to them different disclosure privileges that helps implementing privacy rules. We implemented an information extraction system that finds pairs of attribute-value along with the ownership and an inference engine that given the extracted information and domain knowledge do the reasoning part to detect privacy breaches.

The ontology gives a formal description of the bits of information that might be involved in an information

[1] http://www.vericept.com

[2] A role is a job function or title which defines an authority level.

| Percentage | Emails about marks |
|---|---:|
| Repetitions | 0.03% |
| Mistypings | 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”.

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.

In the following sections, we describe our approach, the important design decisions that we have made developing the PEEP architecture, and the early empirical evaluation of our solution. Section 2 describes the data used in this project. The four component-system architecture is presented 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 text falls into the category of unstructured text which is neither rigidly formatted nor always composed of grammatical sentences [12]. In some aspects, it is similar to manually transcribed spontaneous speech. There is not always explicit punctuation, and it often contains influences such as repetitions of words and omissions. There are also misspellings and acronyms which need to be translated to the appropriate words. An example of email text is given in Figure 1.

Table 1 shows some statistics on the email corpus we are working with. Percentages of repetitions and misspellings were calculated on 12,134 words and ungrammatical utterances on 667 utterances from 43 emails.

Ungrammatical utterances are those where either the subject, verb or object is missing or misplaced. Figure 1 shows an example of ungrammatical email, where verbs are missing in both utterances of the email body `FirstName1 LastName1 (1234567) 80`.

Our corpus is composed of 94 email threads (205 emails) talking about assignment marks. They are mostly emails between students and professors or between professor and teaching assistants. Most emails are queries about marks as shown in figure 1 that introduce exchanges where in each reply a new information is added while previous information is no longer repeated because it became part of a shared context between the sender and the recipient. A typical example of this situation is shown in figure 2. Hence, for information extraction purposes, we considered each email thread as a single text to keep track of all the information exchanged for example to determine the ownership of the information extracted “`FName1 LName1 3333333 scores 40/40`”.

```
From: <Sender@university.ca>
To: <Recipient1@university.ca>
Cc: <Recipient2@university.ca>
Subject: A4 upgrade
Date: Thu, 17 Apr 2003 13:46:12 -0400

Recipient1

The following student should get 80 on A4,
could you please change it?

FirstName1 LastName1 (1234567) 80

Thanks

Sender
```

Figure 1: An email text about updates of assignment marks. the acronym “A4” refers to assignment four. The email was sent to two recipients : Recipient1 and Recipient2. Recipient1, Recipient1, Sender, FirstName1, LastName1, FirstName2 and LastName2 are person names that we changed for confidentiality reasons. In addition, the student identification numbers were replaced by `1234567`.

## 3 Architecture of the system

The privacy compliance engine is composed of four components. The first component is the domain knowledge and consists of 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 component is a pre-analysis module that extracts sender/recipient information and segments the email body for IE purposes. The third component is the IE system which extracts private information from the email body. The fourth and final component detects privacy breaches by using the information extracted from the email body, the recipient/sender information supplemented with additional information from the database, and a set of pri-

```
Hi
I was wondering since the marks have
been posted again whether you may have
forgotten or not?
I received 40/40 on this test,
I can show it to you if needed.
Thanks,
FName1 LName1
3333333

-------- Reply ----------
From: SFName SLName<SLName@university.ca>
To: RFName RLName<RLName@university.ca>
Subject: Re: Marks?
Hi RFName,
I have have seen FName1's Test1(a).& he
scored 40/40 in that test.
Cheers
SFName
```

Figure 2: Two selected emails from an email thread. In the first email, a student (FName1 LName1) gives his score for Test 1. In the second email, a teaching assistant (SFName SLName) confirms that score to professor (RFName RLName).

vacy rules linking concepts from the domain ontology to classes in the information access ontology. Figure 3 shows the four component architecture of the privacy compliance engine.

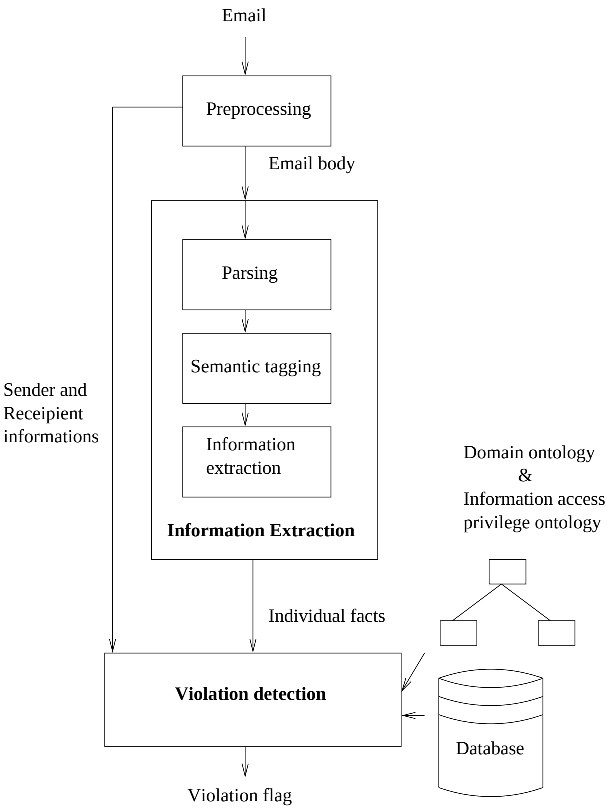

Figure 3: Architecture of the privacy compliance engine.

## 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:

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

   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.
   - an utterance with no punctuation followed by a blank line.

   This segmentation performed on 1239 lines (205 email bodies) produces a segmentation error rate of 2.6%.

## 5 Domain knowledge

The domain knowledge is built on the organization database. It uses knowledge about the roles defined in the database such as *Teacher*, *Student*, *Administrator*, along with a list of table attributes that are potential private information. We distinguish two main knowledge sources: the database which is a core component in any organization, the domain ontology that repre-

| Objects | Attributes |
|---|---|
| Student | student-id,first-name,last-name |
| Staff | staff-id*,first-name,last-name,staff-type |
| Course | course-code*, course-name |
| Department | department-id*,department-name |

Table 2: List of tables and relations of the database.

sents hierarchical relations between roles, and the ontology of the disclosure privileges that associates roles with respective disclosure grants.

### 5.1 The database

To represent domain knowledge about entities cited 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 roles described in the database. There is a direct mapping between the roles of the domain ontology and the ones represented in the database such as student and staff (table 2). However, the database gives additional information related to relations between tables, e.g. "student-registered-course", whereas the ontology describes organizational relations such as "a teacher is a member of the faculty".

This ontology is used by the checking engine to describe privacy rules by linking a particular role to a particular class of disclosure privileges.

In the academic setting our domain ontology is a three layer tree where the person class gathers every physical entity in the database such as *Student*, *Staff* and *Faculty*.

### 5.3 Information access privilege ontology

The information access privileges ontology is a hierarchical organization of the table attributes with a role based taxonomy describing "**who** has the right to release **what**". The role of a person is 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]. Figure 4 shows a part of the information access privilege ontology.

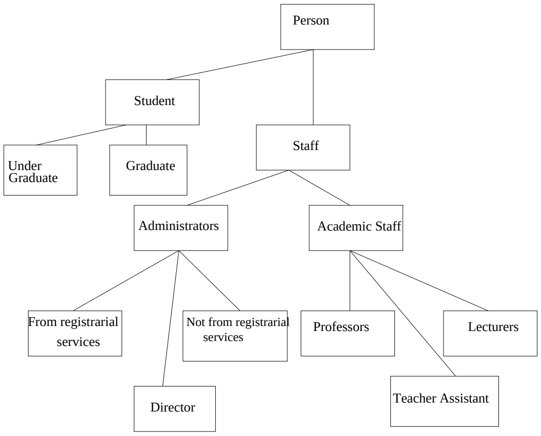

Figure 4: Ontology for the academic domain.

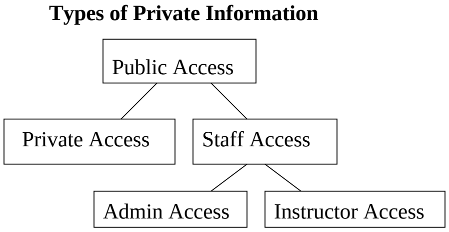

Figure 5: Information access privilege ontology

Each privilege class is composed of a list of attributes that can be released by any individual of the respective role, such as staff versus Staff Access.

## 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 grades.

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 uses a set of word classes to label keywords. The third stage extracts individual facts by first learning patterns and then matching them against a semantically tagged email. The output of the system is a set of relations and facts in Prolog format.

[3] `http://www.cou.on.ca/_bin/publications/onlinePublications.cfm`

[4] `http://www.uoguelph.ca/info/privacyguidelines/`

### 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 [3]. 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) [5]. It annotates keywords like 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", but also named entities such as persons and numbers. Semantic labels are word classes that are determined from the corpus. We used three word classes which are:

- **Verb-Score**, a list of three verbs: `score`, `receive` and `mark`. Other verbs were tagged with a special tag "*Predicate*".
- **Assignment**, a list of keywords of type `assignment X` where `X` is a number and other keywords such as `test` and `exam`.
- **id-number**, a list of keywords such as `identification number` and `ID`.

#### 6.2.1 Approach

This process takes every chunk provided by the parser and looks for a match between the head of the chunk and a keyword. The match is based on the word and its part-of-speech. When a match succeeds, the semantic tag assigned is the word class of the keyword matched. Then, the semantic tag of the head 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 3.

The Fscore of the semantic tagger is comparable to those reported in the proceeding of the seventh Message Understanding Conference (MUC7) [4] for the NE task which was about 97%. However the major source of errors occurs with numbers referring to marks[5] because of the different formats they came in (for example 30/40 as opposed to 30-40 or 30).

| Words | Rec. | Prec. | Fscore |
|---|---|---|---|
| 3978 | 95,0% | 94,4% | 95,5% |

Table 3: Recall (Rec.), precision (Prec.) and Fscore of the semantic tagger.

### 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 semantic role. This stage allows 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.
- The syntactic parsing of ungrammatical sentences generates, usually, partial parses from which it is difficult to infer semantic roles directly.
- 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 [8].

#### 6.3.1 Experiment and results

We annotated 85 sequences of semantic tags generated by the semantic tagger with the semantic roles. We trained first and second order Markov models and evaluated the learning process using the "leave one out" cross validation method [11]. Table 4 shows the average of the recall, precision and Fscore for each model.

The second order model produces better results than the first order model. The analysis of the output shows that classification errors occur when there was no informative context available, e.g. missing keywords such as the verbs `receive` and `get` or nouns such as `mark` or `assignment`. Most common errors were with numbers being identified either as course codes when

[5] The major source of errors reported in the MUC proceedings are proper nouns.

| Model | Rec. | Prec. | Fscore |
|---|---|---|---|
| First order | 68% | 64% | 67% |
| Second order | 73% | 71% | 72% |

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

| Input | Relations | Rec. | Prec. | Fscore |
|---|---|---|---|---|
| Manual | 153 | 37.9% | 51.3% | 46.3% |
| Automatic | 85 | 68.2% | 51.3% | 58,5% |

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

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 reduce those errors. For example some format information such as “a student identification number is a seven digits string” would simplify the detection of student ID’s.

Another source of errors was the use of a particular tag to label irrelevant verbs in the email. This choice explains in part why the second order Markov model outperformed the first order one. Finally, we trained the Markov models on sequences of semantic labels generated at the sentence level: this is the way to learn patterns and to do the pattern matching in standard IE approach. However, since sentence boundaries in emails are not always identified, bits of information may have been missing even after the segmentation was done in the preprocessing stage. We believe that learning on sequences of semantic labels generated at the level of the body of the email can help resolve this problem.

### 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 to avoid over-fitting and to be integrated into the model of the privacy compliance system. Since the overall privacy compliance system was developed in Prolog, we translate each fact and relation into Prolog predicates as shown in figure 1.

The evaluation was made for the 94 email threads and the results are shown in table 5. 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 the 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 final 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 pre-processing component. The engine matches the set of facts and relations against a list of privacy rules 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 information disclosure and are designed to prove a valid information release 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 it 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 domain ontology and the database.

2. The second stage uses the information generated by the first stage to check the disclosure 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 6 shows the different stages of the privacy checking process on an email talking about upgrading a mark. This email was intended to be addressed to the teaching assistant of the course. However the recipient listed is not a teaching assistant. Therefore, the system fails to trigger a privacy rule involving the actual recipient role and the type of information released. In the rule shown above, the left argument (`person`) is the domain ontology class of the recipient. By default, he is granted a public access privilege to

```
From: <Sender@cis.university.ca>
To: <Recipient1@cis.university.ca>
Cc: <Recipient3@cis.university.ca>
Subject: A4 upgrade
Date: Thu, 17 Apr 2003 13:46:12 -0400

Recipient1
The following student should get 80 on A4,
could you please change it?
FirstName1 LastName1 (1234567) 80
Thanks
Sender
```

↓

**Information extraction stage**

↓

```
sender([teacher(['SenderFName','SenderLName']),
        email(['Sender@university.ca'])]).
recipient(1,[person(['Recipient1FName',
                     'Recipient1LName']),
         email(['Recipient1@university.ca'])]).

mark-student([person(['student']),
              mark('80')]).
mark-student([person(['FName1','LName1']),
              mark('80')]).
```

↓

**Checking privacy breaches**

↓

```
Grade Release Denied...!!
Recipient1@university.ca has public-access.
```

Figure 6: Stages of the the privacy checking engine.

information. However, the body of the email identifies the students' names and grades, therefore generating a "`Grade Release Denied`".

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 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.

| Email class | Number | Rec. | Prec. | Fscore |
|---|---|---|---|---|
| Violation | 15 | 20.0% | 20.0% | 20.0% |
| No violation | 79 | 72.1% | 89.1% | 79.8% |
| All threads | 94 | 63.8% | 75.9% | 69.3% |

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

### 7.2 Experiments and results

In order to evaluate the overall privacy compliance engine, we changed the type of some recipients to violate disclosure privileges randomly. We experimented with 94 email threads that had been processed by the IE system. Fifteen of them are considered violation of privacy. The evaluation includes precision and recall for each class of emails. Recall is an indicator of the robustness of the checking engine. Typical cases where the engine fails during the processing is when the engine processes the value of an attribute which has an unexpected format, such as the attribute *Identification number* having the value `id(['mark','change','1597904'])`. These problems are related to the output of the information extraction system.

Table 6 shows the precision, recall, and F-score of the whole privacy checking system.

The privacy checking system produces low results for those emails considered as violations. An analysis of the inference process shows that most errors occurred when the IE extracts incomplete relations. Hence private information couldn't be attached to a particular person and the access privilege defaults to public.

## 8 Conclusion and Future work

In this paper we tackled the privacy checking problem using domain knowledge ontologies and IE techniques. The domain and information disclosure privilege ontologies are formal descriptions of bits of information involved in different privacy violation scenarios. The IE system provides a detailed description of the private information released in an email.

Using an ontology to model domain knowledge and constraints is consistent with the EPAL[6] approach, which makes an ontology a prerequisite for the representation of privacy rules.

[6] `http://www.nwfusion.com`

We addressed the pattern learning in a different way from related works [10, 12]. For example, Soderland used semantic classes to learn regular expressions from on-line rental ads [12]. His system extracted individual facts with an Fscore around 94%. However, the ads are shorter texts with a more restricted format than our emails. The work closest to ours was developed for the CALO project [7], which aims to extract information about people and other entities such as person names, job titles and addresses. They used a conditional random field model [6] to learn Markov models to extract these informations. On emails from the Enron corpus [9], they achieved an average F-score of 80.8%.

Even though the results achieved by the overall privacy compliance engine are encouraging, substantial work can be done to improve our results. For example, we worked on email threads to keep track of the information exchanged in previous emails. Consequently, the IE system generated too much information that misleads the checking engine and decreased its performance. In future work we plan to model email threads in a tree like organization to be able to process emails separately while ensuring easy access to relevant information released in previous emails in the thread.

Another improvement is the use of format information to detect *Student identifier number* and *Course code.* This pre-tagging improved the results of the semantic tagger.

As a long term goal, we plan to tackle two issues. The first one 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 to apply our system to health care domains. We are collaborating with The Ottawa Hospital (TOH) on this application of research.

## 9 Acknowledgements

The authors acknowledge support of the Natural Sciences and Engineering Research Council of Canada and of the Communications and Information Technology Ontario, a division of the Ontario Centres of Excellence, Inc.

## References

[1] S. Abney. Partial parsing via finite-state cascades. *Natural Language Engineering*, 2(4):337–344, 1996.

[2] Q. Armour, W. Elazmeh, N. El-Kadri, N. Japkowvics, and S. Matwin. Privacy Compliance Enforcement in Email. In *Proceedings of the 17th Conference of the Canadian Society for Computational Studies of Intelligence*, Victoria, Canada, 2005.

[3] E. Brill. A simple rule-based part-of-speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, Trento, Italie, 1992.

[4] N. Chinchor and G. Dungca. Four Scores and Seven Years Ago: The Scoring Method for MUC-6. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, pages 33–38. Morgan Kaufmann Publishers, 1995.

[5] N. Chincor, P. Robinson, and E. Brown. HUB-4 Named Entity Task Definition Version 4.8. Technical report, 1998.

[6] A. Culotta, R. Bekkerman, and A. McCallum. Extracting social networks and contact information from email and the web. In *Proceedings of the First Conference on Email and Anti-Spam (CEAS)*, 2004.

[7] D. Richard Kuhn David F. Ferraiolo and Ramaswamy Chandramouli. *Role-based Access Control.* Artech House, Computer Security Series, 2003.

[8] R. Grishman. Information extraction and speech recognition. In *Proceedings of the DARPA Broadcast Transcription and Understanding Workshop*, Lansdowne, Virginie, February 1998. Morgan Kaufmann Publishers.

[9] B. Klimt and Y. Yang. The Enron Corpus: A New Dataset for Email Classification Research. In *In proceedings of the European Conference on Machine Learning*, Pisa, Italy, 2004.

[10] C. D. Manning. Rethinking text segmentation models: An information extraction case study. Technical report, University of Sydney, 1998.

[11] H. Ney, U. Essen, and R. Kneser. On the Estimation of 'Small' Probabilities by Leaving-One-Out. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1202–1212, 1995.

[12] S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 44(1-3):233–272, 1998.

[7] `http://www.ai.sri.com/project/CALO`