Université de Montréal

An Anonymizable Entity Finder in Judicial Decisions

par
Farzaneh Kazemi

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en informatique

Mai, 2008

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé:

An Anonymizable Entity Finder in Judicial Decisions

présenté par:

Farzaneh Kazemi

a été évalué par un jury composé des personnes suivantes:

Jian-Yan Nie,	président-rapporteur
Guy Lapalme,	directeur de recherche
Daniel Poulin,	membre du jury

Mémoire accepté le: .....

RÉSUMÉ

À l'ère de l'information, il y a un besoin croissant de diffuser et partager des données textuelles. Pourtant, quand les données contiennent de l'information sensible et personnelle, la vie privée ne peut seulement être garantie que si les données sensibles sont anonymisées, ou désidentifiées avant leur diffusion.

L'anonymisation est le processus de modifier ou d'enlever l'information d'identification d'un texte de façon à ce que l'individu reste anonyme. Ce processus comporte deux étapes. L'information qui devrait être anonymisé doit d'abord être identifiée, et deuxièmement, elle doit être enlevée, remplacée ou cachée. Cette thèse porte sur la première étape, la détection des données anonymisables.

Un domaine où la nécessité de protéger l'intimité est particulièrement aiguë est dans le système de judiciaire où la plupart des documents contiennent l'information personnelle confidentielle, dont plusieurs requièrent l'anonymisation de données. Cette thèse démontre que les méthodes d'apprentissage automatique peuvent aider en détectant les entités anonymisables dans le domaine de la justice. Un système, appelé Anonymizable Entity Finder (AEF) est construit. AEF emploie une approche d'apprentissage automatique supervisée pour classifier les entités d'un document en deux classes : anonymisable et non-anonymisable. Puisque la plupart de l'information personnelle est des entités nommées, nous nous sommes concentrée sur l'extraction d'entités nommées. AEF emploie le modèle d'entropie maximum comme méthode d'apprentissage de classification parce que ce modèle a obtenu une bonne performance sur plusieurs travaux en traitement de la langue naturelle.

Notre travail est la première recherche sur la détection d'entités anonymisable dans le domaine de la justice. Nos expériences démontrent qu'AEF est un système prometteur pour faciliter le processus d'anonymisation.

Mots clés: Anonymisation, Reconnaissance des Entité Nommées , Désidentification, Décisions de Justice, Entropie Maximum.

ABSTRACT

In the Information Age, there is an increasing need to release and share textual data. However, when data contains sensitive or personal information, privacy can only be guaranteed if the sensitive data is anonymized, or de-identified, before its dissemination.

Anonymization is the process of modifying or removing identifying information from a text so that the individual remains anonymous. Anonymizing personal information within a text involves two steps. The information that should be anonymized must first be identified, and secondly, it must be removed, replaced or concealed. This thesis concerns itself with the first step, that of detecting anonymizable data.

One domain where the need to protect privacy is especially acute is in the justice system, where most documents contain confidential personal information and thus require data anonymization. This thesis demonstrates that machine learning methods can help in detecting anonymizable entities in justice domain. A system, named Anonymizable Entity Finder (AEF) is built. AEF uses a supervised machine learning approach for classifying the entities of a document into two classes: Anonymizable and Non-anonymizable. Since most personal information is named entities, we focused on the Named Entity Recognition. AEF uses the Maximum Entropy model as a classification learning method because this model has achieved high performance in several Natural Language Processing tasks.

This is the first research on detecting anonymizable entities in justice domain. Our experiments demonstrate that AEF is a promising system to facilitate the anonymization process.

Keywords: Anonymization, Named Entity Recognition, De-identification, Judicial Decisions, Maximum Entropy.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

AE	Anonymizable Entity
AEF	Anonymizable Entity Finder
CRF	Conditional Random Field
GIS	Generalized Iterative Scaling
HMM	Hidden Markov Model
NER	Named Entity Recognition
NLP	Natural Language Processing
Non-AE	Non-Anonymizable Entity
POS	Part Of Speech
SFFS	Sequential Forward Floating Search

To my lovely family, Bahman, Sahar, and Sajjad.

ACKNOWLEDGMENTS

First, I would like to express my sincere and deep gratitude to my supervisor, professor Guy Lapalme, for his important support, for his great patience in reviewing my thesis, and for his useful suggestions on my research.

I would also thank the other members of my committee for their feedback and input on this thesis.

I wish to especially thank Mr. Frédéric Pelletier of LEXUM for assisting me in understanding the current annonymization process of judicial documents and providing us sample documents.

My sincere thanks are due to the members of the Recherche Appliquée en Linguistique Informatique (RALI) laboratory, especially Dr. Philippe Langlais and Dr. Jian-Yun Nie for the knowledge that I gained from their courses, and Mr. Elliott Macklovitch for providing a warm research environment in the RALI lab. I warmly thank Mr. Fabrizio Gotti for his voluntarily and friendly help.

I wish to thank all my friends for discussion about the work. Special thank goes to the software developers of OpenNLP Maxent package for their great job.

I would express my warmest gratitude to my parents. Most importantly, I thank my husband Bahman Zamani, my daughter Sahar, and my son Sajjad, for supporting me with their love throughout the entire process of my research. Without their encouragement and understanding it would have been impossible for me to finish this work.

The financial support of the Fonds Québécois de la Recherche sur la Nature et les Technologies (FQRNT) and the University of Montreal are gratefully acknowledged.

CHAPTER 1

INTRODUCTION

Many organizations such as financial firms, medical centers, public health agencies, and statistical institutions collect data that contains personal information and release those data for statistical analysis, scientific researches, and other studies.

For instance, the progress of research in medicine depends on the accessibility and quality of medical databases which include explicit personal health information. Some examples of such information are as follows: first and last names of patients, doctors' first and last names, identification numbers, telephone, fax, pager numbers, hospital names, geographic locations, and dates [26]. However, the publication of personal identity information sometimes causes problems for persons whose information is released. For example, female patients who have had abortions are in peril by anti-abortion groups when their identities are published.

Therefore, dissemination of original collected data and privacy protection are in conflict. To solve this problem, data must be de-identified or anonymized before publication.

There is no consensus on the definition of *anonymization* or *de-identification* in the literature. We choose the following definitions presented in [24].

Anonymization is the process of modifying or removing implicit and explicit identities of a person such that the individual cannot be identified.

De-identification is the process of modifying or removing all explicit identities of a person such as name, address, and phone number.

Obviously, de-identification provides no guarantee of anonymization, since the released information often contains other data that can be linked or inferred to re-identify individuals. As it is mentioned in [24], "Evidence is provided ... that ... [de-identification] process is not sufficient to render data anonymous because combinations of attributes often combine uniquely to re-identify individuals."

In general, both concepts imply the process of concealing personal information in data collection to prevent the identification of the individuals. Definitely, anonymization is more potent than de-identification. *Our aim in this research is to determine the entities which should be anonymized.* As distinguishing between anonymization or de-identification per se is not our interest, we will use the term ***anonymization*** in this text. There are many situations in which these two concepts are used interchangeably, e.g., "many policies, regulations, and legislations in the United States equate de-identified data and anonymous data" [24].

This chapter consists of five sections. Section 1.1 gives the background of the problem and privacy protection in different type of data. In section 1.2, we talk about privacy protection in justice domain and the protocol of personal information protection in judgments. Our motivation for this research is proposed in sections 1.3. Section 1.4 presents the evaluation metrics which are used in our thesis. Section 1.5 is the conclusion of this chapter.

1.1 Background

"Privacy is the ability of an individual or a group to keep their lives and personal affairs out of public view, or to control the flow of information about themselves. Privacy is sometimes related to anonymity although it is often most highly valued by people who are publicly known. Privacy can be seen as an aspect of security" [29]. Privacy limits information sharing and discourages data collecting.

In the Information Age, most collected data (in form of database or document) use sensitive and personal information. With the advances in technology and the increasing use of digital data, privacy is becoming enormously important. The problem is that most data are vulnerable by attackers. For instance, identity theft and blackmail are serious risks for persons whose information subsists in data.

Generally privacy legislation in Canada provides a right of access to information with specific obligations to protect the privacy of individuals by restricting the collection, use, and disclosure of information about those individuals [21]. Removing personal

information is the main goal of the de-identification or anonymization process in which the data remains useful in accordance with legislation.

1.1.1 Privacy protection in database

The problem of disclosing individuals' data such that their identities cannot be identified is not a new problem. There are much related works in the statistical and medical domains. Statistical agencies are often dealing with personal information and they should protect the individual's privacy for applications such as data mining, cost analysis, fraud detection, and retrospective research [25]. The statistical data are normally stored in database tables containing records, each record including some fields that contain the individual's information. Each field has a name which is already determined at the database design time. Looking at the database schema, it is not difficult to recognize the important identities that should be anonymized. For instance, table 1.1.1 shows a schema of an Employee table in a database.

Employee ID	**First Name**	**Last Name**	**Date of birth**	**Sex**	**SIN**	**Salary**

Table 1.1: The Schema of Employee Table in a Database

It is obvious that a field in the table corresponds to a specific information. For instance, if we are asked to anonymize the first name, we see that the first name is the second field in this table, hence, doing anonymization is a straightforward task. Indeed, the problem here is how to anonymize fields such that an attacker cannot identify an individual from publicly available information by linking or combining the data.

In order to protect an individual's information, various methods can be applied in different domains for concealing identities. The following methods taken from [24] are examples.

Suppression: The sensitive data are not released. Therefore, the quality of information is reduced and the rendered data sometimes is useless.

Substitution: The sensitive data are replaced with another data in its equivalence class. For example, replace a real name by a fictive one.

Generalization: Data is replaced with a more general case. For instance, replace an address with the corresponding province name.

Additive noise: Additive noise involves the random incrementing or decrementing of data while keeping their aggregate values similar.

Encryption: The conversion of data into secret form by means of a secret key known only to people who are allowed to see the details.

1.1.2 Privacy protection in text documents

In the past, released information was mostly in database format; however, today it is disseminated in other forms such as text or web pages. In the medical domain, both database and plain text are released. In United States, the Health Information Portability and Accountability Act (HIPAA) provides a list of Personal Health Information (PHI) which should be removed from medical documents or database for de-identification [26].

The problem of anonymization or de-identification in documents is more complicated compared to database. In a plain text, we should firstly determine the entities which are related to personal information and those that should be anonymized, while in a database, entities about personal information are already determined by the database schema.

In the judicial domain, made decisions are a source of law according to the common law in Canada. Therefore, the judicial decisions are a prominent source of law for courts, lawyers, and public. Some judicial decisions are edited before disclosure to ensure compliance with publication ban and privacy rule. For example, family law matters are particularly sensitive and they must be modified before publication in some provinces. These types of judicial decisions should be anonymized in such a way that the privacy of participants in decisions is protected while the documents still remain understandable by the public.

The header of a judicial decision contains the name of parties (e.g., defendant and plaintiff) which are generally anonymized, but there are other information in the body of the document, such as names of relatives, that should be anonymized too.

1.2 The case of judicial documents

In the past, court judicial decisions were accessible to the public through law libraries, court registries, and legal publishers. The print media used then does not lend itself to data mining and therefore, there were more control over the dissemination of the decisions. As a result, privacy protection was not a big issue. Nowadays, these decisions are available over the Internet. In Canada, Canadian Legal Information Institute (CANLII) [1] provides a free legal access to all Canadian jurisdiction decisions.

Publication of judicial decisions on the web is an opportunity for the public to understand how court decisions are made, also help people and lawyers to be familiar with the law and different judicial decisions. In addition, the availability of these decisions ensure people of the openness of justice, especially in common law legal systems. Furthermore, free access to all decisions facilitates research for the legal profession, the media, and the public.

The number of court documents is huge. For instance, the number of decisions made only by tribunal judiciary is estimated by [16] at about 200,000 decisions annually, which represents 2,000,000 text pages. Due to privacy protection, not all such documents can be widely distributed with original form. Therefore, there are restrictions on the publication of certain personal information disclosed in the decisions. Some court decisions are edited before publishing, to comply with privacy rules for protecting persons who are participants in a judicial procedure. According to privacy rules, any information that leads to identify a certain person should remain confidential [16].

Consequently, the justice decisions should be anonymized before their publication. However, anonymized documents should remain understandable for public. That means, the documents are still readable and useful even after having removed or modified some personal information. When done manually, an editor must peruse a document to determine the entities which should be anonymized. Due to the voluminous data in this domain, this anonymization process is tedious, requiring one or two minutes on average per a text page [16].

[1] http://www.canlii.org/

NOME [15] is an assistant application which reduces the processing time by highlighting potential proper names in a document. However, the number of determined proper names is much more than the number of anonymizable proper names. Therefore, the editor must filter the list of suggested proper names and select the anonymizable ones. This process still takes a long time when the proper names list is long.

In this thesis, we describe the development of a system using machine learning technique which determines the anonymizable proper names in order to reduce the humane filtering time. A motivating example is shown in section 1.3.

1.2.1 Disclosure of personal information in judicial decisions

Canada's court system involves four levels of hierarchy (figure 1.1) [13]. The highest level of the system is Supreme Court of Canada which has jurisdiction over all courts. The courts of appeal are the next level of system such as the Federal Court of Appeal, the provincial courts of appeal, and Martial Court of Appeal. These courts of appeal hear cases which appealed from the Federal Court, provincial superior courts, or military courts.

The third level includes the provincial/territorial superior courts (sometimes called Supreme Court in some provinces) and the Federal Court. Provincial/territorial superior courts exercise a trial jurisdiction on a variety of issues as in family and in important civil or criminal matters. Superior courts also exercise judicial control over Federal court which deals with the matters specified in federal statutes such as immigration. The Tax court of Canada and military courts are specialized courts created in order to deal more effectively with certain cases. The lowest level is provincial/territorial courts which deal with lesser cases whether criminal or civil and youths.

Therefore, there are different types of judicial decisions. The decisions involving family, and youth matters are the most sensitive to publish over the Internet. Some courts do not distribute these decisions. In order to publish all decisions, courts need anonymization processing.

Our data is chosen form decisions of 2002 of two Superior courts; Superior Court of Ontario and Supreme Court of British Columbia.

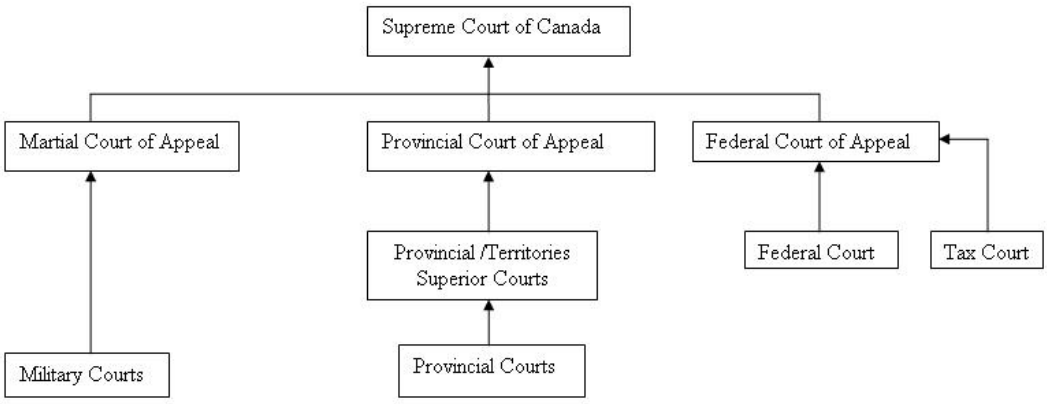

Figure 1.1: Chart of Different Levels of Courts (adapted from [13])

Courts across Canada use several solutions to protect the privacy of parties and others involved in litigation, e.g., removing personal information or using initials instead of the person name. Removing person names from a document is not sufficient, indeed, a study has shown that 87% (216 millions of 248 millions) of the population in the United States can be uniquely identified using ZIP code, gender, and date of birth [24]. Legislation determines what kind of personal information should be concealed from the public.

Judicial council of Canada considers the following levels of protection [7] :

1. **Personal Data Identifiers**

2. **Legal Prohibitions on Publication**

3. **Discretionary Protection of Privacy Rights**

The Canadian Judicial Council [7] determines the specific type of information requiring to be protected.

1. **Personal Data Identifiers:** Table 1.2 shows a list of personal data identifiers according to [7]. Individuals have the right to the privacy of this information and it should be omitted from all decisions. This information when connected with a person's name could identify the person. This type of information is rarely used in court documents except birth date.

Day and month of birth
Social Insurance Numbers
Credit card numbers
Financial account numbers

Table 1.2: List of Personal Data Identifiers

2. **Legal Prohibitions on Publication:** Certain participants in the judicial proceeding are subject to a statutory or common law restriction on publication. In Canada, Youth Criminal Justice Act matters, criminal jury matters, sexual or violent criminal matters have the most common bans in their context. These bans prohibit the publication of the identity and any information which would reveal the identity of a complainant, a witness, or a youth. However, removing a person's name who should be protected by a publication ban is not sufficient to forbid disclosure of identity. It is possible that other information connected to an individual helps identify an individual. Accordingly, further information should be anonymized to avoid disclosure of identity.

   According to [7], three types of information have to be protected:

   Personal Data (table 1.3) contains a list of personal data allowing the identification of a person directly or indirectly that should be anonymized when there is a publication ban.

Names, nicknames, aliases
Day and month of birth
Birthplace
Addresses: street name and number, municipality, postal code, phone, fax, e-mail, URL, IP address Unique personal identifiers (e.g., numbers, images or codes for social security, health insurance, medical record, passport, bank or credit card accounts)
Personal possession identifiers (e.g., license or serial number, property or land identification, corporate or business name)

Table 1.3: List of Personal Data

Personal Acquaintances Information (table 1.4) includes names and personal data of persons or organizations related to an individual whose identity must be anonymized.

Extended family members: parents, children, brothers, and sisters, in-laws, grandparents, cousins
Foster family members, tutors, guardians, teachers, babysitters
Friends, co-habiting persons, lessors, tenants, neighbors
Employers, employees, co-workers, business associates, schools, sports teams

Table 1.4: List of Personal Acquaintances Information

Specific Factual Information (table 1.5) is the information that can increase the risk of identification. Even if the personal data and personal acquaintances are concealed from the judgment, there is still a minimal risk of identification through Specific Factual Information .

Names of communities or geographic locations
Names of accused or co-accused persons (if not already included in the publication restriction)
Names of persons acting in an official capacity (e.g., expert witnesses, social workers, police officers, physicians)
Extraordinary or atypical information on a person (e.g., renowned professional athlete, very large number of children in the family, unusually high income, celebrity)

Table 1.5: List of Specific Factual Information

3. **Discretionary Protection of Privacy Rights:** other personal information should be omitted if the dissemination of this information could harm innocent persons, minor children, third parties, or subvert the course of justice. When there is no publication ban, protection of the innocent from unnecessary harm is a valid and important policy consideration. In these cases, the judge must balance this consideration with the open court principle by asking how much information must be

included in the judgment to ensure that the public will understand the decision that has been made [7].

In summary, the following information should be anonymized from judgments:

- Personal data identifiers of all individuals.
- If there is a publication ban for an individual
  - Personal data of the individual.
  - All information about relatives of the individual.
  - Other information that can help to identification of the individual.
- Personal information about innocents.

Hence, the majority of personal information in decisions that should be anonymized are names (e.g., parties name or relative name), day and month of birth, and address. The birth year of a person is intact.

1.3 Motivation

Due to privacy issues, there are limitations on the dissemination of personal information. Therefore, some documents must be anonymized before publishing because personal information of participants in a decision procedure should remain confidential. As it was pointed out in the previous section, the information about individuals in decisions which should be anonymized are generally names (e.g., person name, closed relative names), location (address), date (birth date), number (e.g., phone number, pager number) which are mostly Named Entities.

Since most of the anonymizable information are named entities (person name, birth date, address) therefore, detecting named entities in a text is a first step for doing anonymization. Extracting named entities from text usually is known as Named Entity Recognition (NER). NER is a subtask of information extraction which detects and classifies the named entities such as name of persons, organizations, locations, time, and date. Many

Natural Language Processing (NLP) applications need to find named entities in the text documents and many approaches are used for the recognition of named entities. The Conference on Natural Language Learning (CoNLL) had a shared task on named entity recognition in 2002 and 2003.

Original version	**Anonymized version**
The parties are the parents of an infant child,William Millar. William was born on March 17, 2000. Ms. Green was represented by Ms. Bond at the trial, and Mr. Millar represented himself. Ms. Green purchased a home in her sole name at #3 - 230 East Keith Road, North Vancouver. Mr. Millar worked under the business name "MetroGnome PC Systems".	The parties are the parents of an infant child, W.M.. W.M. was born on [. . .], 2000. Ms. J.L.G. was represented by Ms. Bond at the trial, and Mr. D.W.M. represented himself. Ms. J.L.G. purchased a home in her sole name at [. . .], North Vancouver. Mr. D.W.M. worked under the business name "M.[. . .] Ltd."
Ms. Macdonald's mother holds the voting shares in Indian River. Since 1990 and throughout the marriage Susan MacDonald received monies from Indian River. MacDonald is a Vice-President in the Capital Market Division of R.B.C. Dominion Securities. The loan accounts related to the RBC Dominion Securities investment accounts are acknowledged to be family debts.	Ms. S.R.M.'s mother holds the voting shares in I.[. . .] Ltd. Since 1990 and throughout the marriage S.R.M. received monies from I.[. . .] Ltd. I.A.M. is a Vice-President in the Capital Market Division of R.[. . .] Securities. The loan accounts related to the R.[. . .] Securities investment accounts are acknowledged to be family debts.

Table 1.6: Examples of Anonymizable Entities in sentences from two decisions. Entities that should be anonymized are underlined in the left column. The right column shows the anonymized version of same sentences in which underlined strings indicate the replacement of corresponding anonymized entities.

Table 1.6 shows some examples which are selected from the current anonymization task that is based on using NOME and human finalizing. These examples illustrate some of the complexities of anonymization. There are different types of entities that

should be anonymized, such as (last or first) names, birth dates, address, and organization names. The information of an individual that should be anonymized is replaced with initials or with an omission mark between square brackets. Initials are used for person's name; One initial for last name, one for first name, and one for middle name without space. E.g., "William Millar" is replaced with "W.M." (table 1.6). The information that should be removed from a document, is replaced with omission mark between square brackets "**[...]**". For instance, the birth date of William (March 17) is replaced with **[...]**. The name of an organization that should be anonymized must be replaced with its first initial followed by omission mark between square brackets. E.g., R.B.C. Dominion is replaced with R.[...]. Since many individuals may have same initials, a number is added immediately after initials. For instance, if there are two persons with the same initial (e.g., S.R.) in a document that should be anonymized, the name of first person is replaced with S.R.1 and the second with S.R.2 [14].

As we see, some anonymized entities are a common noun such as *"Green"*. Moreover, there are proper names that should not be anonymized such as *"Ms. Bond"* because he is the advocate (Counsel). A more difficult case is to find out "how can we recognize that "Indian River" is a location name or an organization name?"

In a document, the name of a person may appear in different ways. For instance, "William Millar" may be referred to as "William," "Mr. Millar," or "Dr. Millar." All the names that refer to the same person, should be replaced with the same string. Coreference Resolution is the task of determining entities that have the same reference. Coreference resolution of person names can help to identify the variant names of the same person. In addition, extracting the semantic relationships between named entities, e.g., person and her residing place, person and her birthday, person and her organization, that help to detect the information about the same individual is another problem that should be considered.

Finding anonymizable entities is a more complex task than the named entity recognition one. We should not only detect the named entities within a document but also we must find which ones should be anonymized and make sure that all occurrences are replaced by the same string.

RALI[2] and LEXUM[3] have developed the NOME application which determines only potential proper names. The input of this application is a MS-Word document and the output is a list of terms which contains the proper names in which anonymizable terms must be selected. The problem is that this list often contains much noise: it includes many terms which should not be anonymized. The anonymization process still takes a long time for voluminous documents. In this thesis, we explore how the use of machine learning methods help reduce the noise and thus the time for anonymization.

For instance, figure 1.2 shows using NOME for a 20-page decision (appendix I) that contains 8,522 words. The output of NOME (enlarged in figure 1.3) is a list of 54 terms of which only 6 to be anonymized. "Child Support Guidelines," "Divorce Judgement," "Threshold Condition," and "April" are examples of output terms that should not be anonymized. The method used by NOME is simple: every sequence of two or more capitalized words that are not separated by a dot is considered a proper name. From now on, whenever we use the term "capitalized word" that means the first letter of word is capitalized.

Since a decision contains many capitalized words, especially in the headline, the list is long and contains much noise. As shown at the top of figure 1.3, a person name (e.g., "Scantland") can appear many times in the list and the editor must choose the same string for all occurrences.

NOME uses three lists to help detect the proper names. These lists can be modified by the editor.

Exclusion list contains the list of words that are never highlighted by NOME. For example, the names of countries.

Inclusion list includes the list of words which must always be highlighted by NOME. For example, the name of small cities.

Title list contains all titles of persons for detecting person names. For example, "Mr." and "Ms.".

[2]http://rali.iro.umontreal.ca/

[3]http://www.lexum.umontreal.ca/

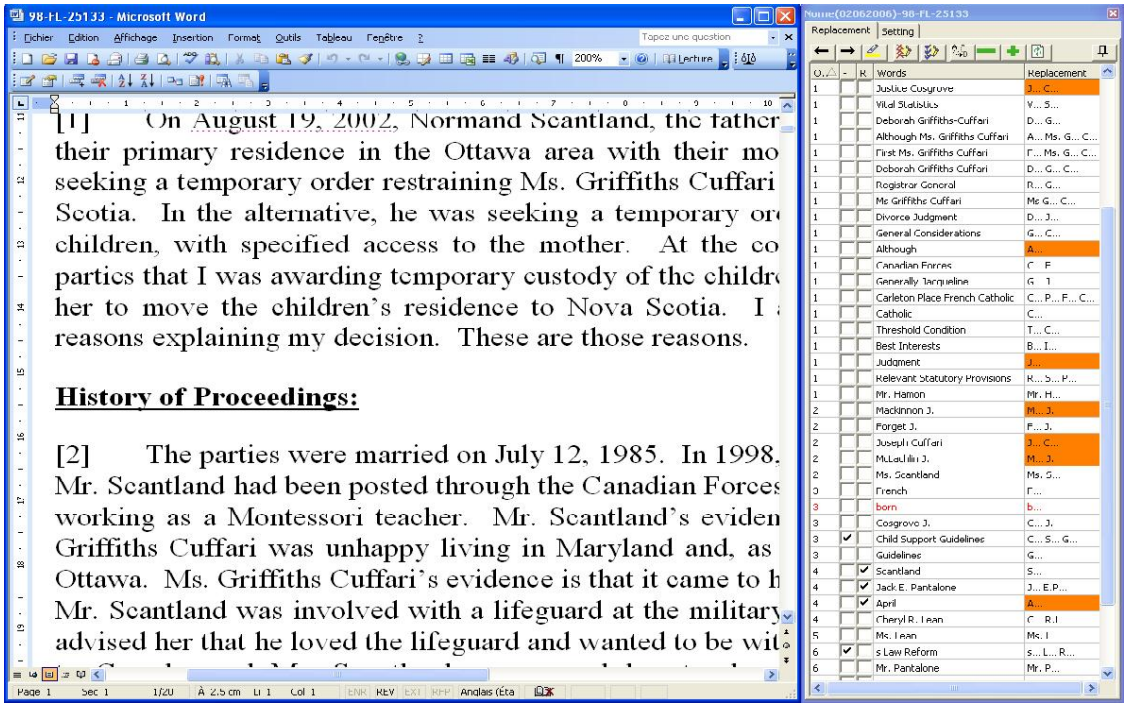

Figure 1.2: A snapshot of NOME. After opening a document with Microsoft Word (left window) and executing the macro NOME, it opens the NOME interface window (right window enlarged in figure 1.3) containing a list of terms. Each row of this list contains four items: number of occurrences of the term in the document, a check box for selecting the term that the editor wants to remove from the list, a check box for selecting the term that should be replaced with the replacement string (initial), the term itself, and the replacement string proposed by NOME. Some replacement string are highlighted because there are two or more proper names that have same initials. The editor can modify the replacement term. After selecting the terms that should be anonymized. The editor selects a command from the buttons at the top of the right window to have NOME change the terms automatically.

Occs	-	R	Words	Replacement
1			Should Mr. Scantland	S... Mr. S...
1			April Mr. Scantland	A... Mr. S...
1			Normand Scantland	N... S...
111			Mr. Scantland	Mr. S...
2			Ms. Scantland	Ms. S...
4			Scantland	S...
4			April	A...
1			Deborah Griffiths-Cuffari	D... G...
1			Although Ms. Griffiths Cuffari	A... Ms. G... C...
1			First Ms. Griffiths Cuffari	F... Ms. G... C...
1			Deborah Griffiths Cuffari	D... G... C...
110			Ms. Griffiths Cuffari	Ms. G... C...
1			Ms Griffiths Cuffari	Ms G... C...
2			Joseph Cuffari	J... C...
14			Mr. Cuffari	Mr. C...
1			Although	A...
1			Canadian Forces	C... F...
2			Mackinnon J.	M... J.
2			McLachlin J.	M... J.
1			Metivier J.	M... J.
3			Cosgrove J.	C... J.
2			Forget J.	F... J.
3			Child Support Guidelines	C... S... G...
3			Guidelines	G...
1			Mr. Hamon	Mr. H...
1			Relevant Statutory Provisions	R... S... P...
6			s Law Reform	s... L... R...
1			Divorce Judgment	D... J...
1			Judgment	J...
9			Divorce	D...
1			Threshold Condition	T... C...
1			Material Change	M... C...
1			Best Interests	B... I...
1			General Considerations	G... C...
1			Generally Jacqueline	G... J...
15			Jacqueline	J...
1			Carleton Place French Catholic	C... P... F... C...
1			Catholic	C...
3			French	F...
1			Registrar General	R... G...
1			Vital Statistics	V... S...
1			Justice Cosgrove	J... C...
1			Exhibit E	E... E...
1			New Year	N... Y...

Figure 1.3: Enlargement of the NOME interface window of figure 1.2 showing noise such as “Child Support Guidelines” and“Divorce Judgement” that NOME produces. The first 6 entities related to “Scantland” must be distinguished by the editor to keep only the ones to be anonymized and replaced by the appropriate string.

The goal of this work is to identify the entities that should be anonymized in decisions. We will be focusing on named entities which are important for anonymization. In the next chapter we will describe the Maximum Entropy model that we use as a machine learning algorithm for extracting anonymizable entities. Feature selection being important in a machine learning algorithm, we will explore the impact of different features on the performance of the model. We will then focus on named entities which should be anonymized and try to detect only anonymizable named entities. In our experiment, we will use the Java-based openNLP maximum entropy package [1]. In order to assess the quality of our results we will use the evaluation metrics described in the next section.

1.4 Evaluation metrics

The performance of a classification system can be evaluated using a confusion matrix [11]. The predicted classes assigned by a system compared with a manual class assignments by an expert. Table 1.4 shows the confusion matrix for a binary classifier, where

True Positives (TP) is the number of examples correctly predicted as positive.

False Positives (FP) is the number of examples incorrectly predicted as positive.

True Negatives (TN) is the number of examples correctly predicted as negative.

False Negatives (FN) is the number of examples incorrectly predicted as negative.

		Predicted	
		Pos.	Neg.
Actual	Pos.	TP	FN
	Neg.	FP	TN

Table 1.7: Confusion Matrix

Given a confusion matrix, a few metrics are commonly defined as basic measurements such as precision and recall. In binary classification, **Precision** is the ratio of the

number of true positives to the number of predicted positive examples , and **Recall** is the ratio of the number of true positives to the number of actual positive examples.

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

F-measure (F_β) is a trade off between precision and recall.

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P+R}$$

The default value for β is 1. F_1-measure is a general measurement in Natural Language domain. The importance of precision and recall are equivalente for this measurement. The F_2-measure is used when recall is more important than precision. In our work, in addition to F_1, we consider also F_2-measure, because a high recall is more important for us. High recall indicates that most anonymizable entities are detected.

The quality of anonymization task is related to how well Anonymizable Entities (AEs) are detected. We assume that the judicial text consists of a set of AEs (Positive class) and Non-AEs (Negative class). An entity could be a word (or token). We redefine above definitions as following:

TP : number of entities that are correctly recognized as AEs.

FP : number of entities which are recognized as AEs but they are not AEs.

TN : number of entities that are correctly recognized as Non-AEs.

FN : number of entities which are recognized as Non-AEs but they are AEs.

$$Precision = \frac{\#correctly\ recognized\ AEs}{\#recognized\ AEs}$$

$$Recall = \frac{\#correctly\ recognized\ AEs}{\#AEs}$$

High recall means that all or almost entities to be detected are determined. Since our goal is detecting the entities in a document which should be anonymized, high recall is important because detecting all entities that should be anonymized is extremely important in decisions.

1.5 Conclusion

Since publication of personal information in database or documents conflicts with privacy protection, they should be anonymized or de-identified before publishing. The problem of anonymization in a database is finding a best model to answer the question "Which fields of a database containing information about individuals, should be anonymized while keeping the data remains useful and ensuring that individuals cannot be re-identified?" In text documents, this problem is more difficult because we should first detect the information about individuals in the documents, while in the database, they are already determined.

In the judicial domain, anonymization is essential because individuals have the right to the privacy of their personal identifier. Moreover, there are some publication bans for certain participants in the judicial process. Three types of information that must be protected are explained in section 1.2. Given the huge number of these documents and their relatively long length, the anonymization task is tedious and costly in human effort.

NOME is an assistant application that highlights potential proper names in a document. An editor must filter the list of suggested proper names. In addition, the editor must seek the whole document for finding other entities that should be anonynmized that NOME has not found. However, the number of suggested proper names is much more than the number of anonymizable proper names. Therefore, this process still takes a long time when the proper names list is long.

Since most of the information about an individual in decisions are named entities such as person name, birth date and address, we focused on the task of extracting named entities. This task is known as Named Entity Recognition (NER). Many machine learning algorithms are applied on the named entity recoition task. Our aim is to apply a

learning algorithm on our problem and explore how machine learning methods help detect anonymizable entities.

In the next section, we introduce the problem of named entity recognition and its main approaches. We present some anonymization systems which are mostly used in the medical domain. In addition, we review some selected works on named entity recognition. Since the maximum entropy model is popular in natural language processing, we introduce this model that will be used as a probabilistic learning algorithm on our problem.

CHAPTER 2

RELATED WORK

Extracting identities from a text can be seen as an application of Named Entity Recognition (NER) because most identities are named entities. In Section 2.1, we present the NER, its main approaches with their advantages and disadvantages. Section 2.2 introduces some anonymization systems which are mostly used in the medical domain. Section 2.3 is an overview of selected works in NER. The mathematical concepts of maximum entropy approach are described in the section 2.4. The conclusion of this chapter is given in section 2.5.

2.1 Named Entity Recognition

NER is a subtask of information extraction which detects and classifies the elements in a text, e.g., name of persons, organizations, locations, time and date.

NER is currently being used in various domains of natural language processing such as text summarization, question answering, cross-language information retrieval, as well as other domains including medicine and bioinformatics.

As an example, the following sentence contains three named entities: *Mr. Scantland* is of type person, *Canada* is of type location, and *August 1998* is of type date.

[**PER** Mr. Scantland] moved back to [**LOC** Canada] in [**Date** August 1998].

NER is not a trivial task. There are ambiguities in the identification of entities. Ambiguity exists between location names and organization names, location names and person names, person names and organization names, or even person, location, and organization names. For example, the name *Sparks* could be either a last name or a geographical location (a city name in Nevada). Sometimes the name of a company is also the name of some of its founders.

The term "Named Entity" was created in the Message Understanding Conference (MUC) in which researchers present their works on different fields of information ex-

traction and develop new and better methods and standards for evaluation. NER was introduced in 1995 by the MUC-6 conference. In addition, CoNLL-2002 and CoNLL-2003 conferences have organized a shared task on Language-Independent Named Entity Recognition.

According to [4], the main approaches to NER are the following, however most systems use a combination of these methods.

Lexical lookup Uses a handcrafted lexicon list. For instance, a reference book containing persons' last names or a Gazetteer, a list of places, organizations and people.

Rule-based Rules are extracted from a corpus to identify named entities. The rules may be structural, contextual, or lexical. For instance, the following rule

$$Title+ \text{ Capitalized word} \rightarrow \text{Title Person-name}$$

is a rule which helps find a person entity in English texts. Regular expressions can help detect entities such as dates and times.

While handcrafted rule based systems achieve a high performance, they have several disadvantages:

- Generating rules is a time consuming work. Producing handcrafted linguistic resources such as context free grammars, regular expressions, lists of trigger words, and gazetteers require a considerable amount of time, a lot of human effort, and a significant computational linguistic knowledge [22]. Therefore, the performance of the system is dependent on the capabilities of the human designer.
- It is difficult to adapt a handcrafted rule based system to other domains or languages because [22]
  - the features of documents are likely to change from one domain to another. Moreover, handcrafted rule-based systems can perform well on a given collection while they will not perform so well on a different one.

  - the rules of a language are likely to change from one language to another.

- Some rules often have many exceptions. Extracting the rules or patterns from a text requires many discourse and linguistic features which are difficult to predict [12].

Statistics-based & Machine learning The high cost of manual rule drafting and knowledge extracting prompted researchers to investigate the application of machine learning approaches [12]. The main idea is to learn from annotated training examples by computational and statistical methods and to find a function or a classifier that can classify the unseen examples. Some machine learning algorithms that are used are as follows: Neural Network, Decision Tree, Hidden Markov Model, and Maximum Entropy.

Several research experiments have shown that learning systems produce good results compared with handcrafted rule-based systems [12]. Machine learning techniques have several advantages:

- Consideration of more contextual features than with handcrafted rules [12].
- The independence of language and domain, provided that there is training data.
- Reduction of human effort because annotation of a document is easier than the extraction of rules [12].
- Learning from former task. The information extracted from a previous task can become the features in an advanced task and help the system to improve.

Machine learning techniques also have disadvantages:

- In supervised learning, we have to prepare training examples. For instance, we should tag each example with a class tag for a classification problem.
- Outside of adding new examples, machine learning methods are harder to tune than rule-based approaches in which it is only a method of adding new rules.

2.2 Anonymization Systems

A lot of work has been done in de-identification or anonymization of clinical documents, however the research dealing with judicial decisions is rare. In this section we will survey some systems and techniques for anonymization problem, most of them being used in the medical domain.

Datafly [24] is an anonymization system for databases. This system utilizes the *k-anonymity* technique which ensures that any individual cannot be distinguished within a group of at least k individuals [25]. *k-anonymity* is a well-known privacy model for structured data. A data set satisfies *k-anonymity* if and only if the minimal set of attributes in a table that can be linked with external information to re-identify individual records appears with at least k occurrences in the same data set. Privacy is better when k is large, but not too large as to make the data useless.

Scrub [24] presented by Sweeny detects explicit personal information in general medical documents. This system utilizes a set of detection algorithms (patterns) competing in parallel to label terms of text as being a proper name, an address, a phone number, and so forth. Furthermore, a host of lists such as lists of common first names, are used for detecting personal information [24]. This system uses a combination of rule-based and lexical lookup approaches with an accuracy of 98-100%.

Name De-identifier using semantic selectional restrictions [27] uses the maximum entropy statistical model for detecting only names in general medical texts. "The proposed algorithm is based on estimating the fitness between candidate patient name references with a set of semantic selectional restrictions. The semantic restrictions place tight contextual requirements upon candidate words in the report text and are determined automatically from a manually tagged corpus of training reports" [27]. All references to patient names and the logical relation between a name and their local contexts in which the names were used are tagged. The maximum entropy model calculates the conditional probability of a logical relation for

a given context [27]. The best overall performance is reported with recall score of 93.9% and precision score of 99.2%.

We will use the same probabilistic model (Maximum Entropy) but we only tag words as an anonymizable or a Non-anonymizable entity.

NOME (see figures 1.2 and 1.3) is an MS-Word macro which assists editors in the anonymization process of judicial decisions. This macro detects potential proper names in a judicial decision and gives the opportunity to the user to automatically replace detected names by their initials or by any other characters [15]. Anonymization is thus a semi-automatic based on the logic structure of NOME. A long list of capitalized words is generated and user has to select the proper names that should be anonymized [16].

The basic logic of NOME is that every sequence of two or more capitalized words (the first letter of the word is capital) that are not separated by a period is considered a proper name. Since all capitalized words are not proper names, e.g., the first word of a sentence, then NOME tries to eliminate the number of selected words by using three lists:

1. An *Inclusion* list that includes the list of words which must always be highlighted by NOME. For example, we can add the word **born** to the inclusion list since the birth date is generally mentioned after this word when detecting a birth date is important.
2. An *Exclusion* list which contains the list of words that are never highlighted by NOME. For example, the names of countries.
3. A *Title* list contains all titles of persons which are generally used in documents to help detect person names.

According to the classification given in the previous section, NOME, combines uses lexical lookup and handcrafted rule-based methods.

One problem with NOME is that according to its rules (a sequence of two or more capitalized word), it cannot detect the single potential proper names and the person

names which are written in all capitalized letters. Moreover, the output list is often large and it takes long time to select anonymizable names.

2.3 Overview of some works in NER

This section refers to some works done in NER. At the CoNLL-2003 conference , different approaches to NER have been compared in a Shared Task. We will now review the algorithms, data, and results of this shared task.

Since maximum entropy model has obtained good results for NER task in CoNLL-2003 Shared Task, we select some of the papers which have used this model in order to investigate the methods they have applied in Named Entity Recognition.

2.3.1 Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition

The CoNLL-2003 Shared Task [28] considers four types of named entities: Persons (PER), Locations (LOC), Organizations (ORG), and Miscellaneous (OUT) (the entities that do not belong to previous three groups). To classify a named entity, one should consider more than four labels since an entity may consist of two or more words. Therefore, some other tags are added to four previous labels to indicate the position of a label. For example X-B label for the beginning of named entity, X-C for the middle of named entity, X-E for the ending of named entity, and X-U for unique named entity. Several approaches of NER are applied to the same data set (English and German) and their performances are compared.

Technical details of the paper are discussed in the following.

Data

The participants have used two languages, English and German, as data. The English data is from Reuters Corpus and the German data is from ECI Multilingual Text Corpus. A tokeniser, Part Of Speech (POS) tagger, and a chunker are applied to the raw data. Named entity tagging of training, development, and test data is done manually.

Algorithms

The results of different learning methods on the same data set are compared in the evaluation of the Shared Task. The performance is measured with F-Measure (see section 1.4). Sixteen systems have participated in this task with a wide variety of machine learning techniques as well as system combinations. Some systems have used additional information like a Gazetteer or unannotated data. All participants except one have used lexical features. Most of the systems have used POS tags. Eleven of the sixteen systems have attempted to use additional information in addition to the given data.

The most frequently applied technique in the CoNLL-2003 Shared Task is Maximum Entropy Model (MEM). Statistical learning method, Hidden Markov Model (HMM), Conditional Markov Model (CMM), Support Vector Machine (SVM), Conditional Random Field (CRF), Transformation-Based learning, Memory-Based Learning, Voted Perceptron, and AdaBoost are other methods which were used in the Shared Task.

Results

The results are shown in table 2.1. The top three results have used the maximum entropy model. Therefore, it seems that maximum entropy is a good choice for this kind of task. Also a combination of different learning systems has proved to be a good method for obtaining high results. Florian et al. and Klein et al. have tested different approaches for combination of methods.

Generally the systems which used a Gazetteer seem to benefit more than others which used unannotated data. But the results of Zhang and Johnson show that there is no difference between using Gazetteer and unannotated data [28].

Following section introduces two of the papers that have used maximum entropy in their experiments and have gained good result.

2.3.2 Related Papers

Chieu and Ng [6] present a NER with maximum entropy approach (explained in next section) using local and global features to classify each word [6], [5]. Local features of a

System	Technique	%Precision	%Recall	%F
Florian	A combination of four classifier: Hidden Markov Model, Maximum Entropy, Transformation Based Learning, Robust Risk Minimization	88.99	88.54	88.76±0.7
Chieu	Maximum Entropy	88.12	88.51	88.31±0.7
Klein	Character-level Hidden Markov Model and Maximum Entropy Markov Model	85.93	86.21	86.07±0.8
Zhang	Robust Risk Minimization	86.13	84.88	85.50±0.9
Carreras(b)	AdaBoost	84.05	85.96	85.00±0.8
Curran	Maximum Entropy	84.29	85.50	84.89±0.9
Mayfield	Support Vector Machine	84.45	84.90	84.67±1.0
Carreras(a)	Perceptron	85.81	82.84	84.30±0.9
McCallum	Conditional Random Field,	84.52	83.55	84.04±0.9
Bender	Maximum Entropy	84.68	83.18	83.92±1.0
Munro	Character N-Gram Modeling	80.87	84.21	82.50±1.0
Wu	A combination of three classifiers: Transformation Based Learning, Support Vector Machine, Boosting	82.02	81.39	81.70±0.9
Whitelaw	Character-based Probabilistic approach	81.60	78.05	79.78±1.0
Hendrickx	Memory Based Learner	76.33	80.17	78.20±1.0
De Meulder	Memory Based Learner	75.84	78.13	76.97±1.2
Hammerton	Long Short-Term Memory	69.09	53.26	60.15±1.3
baseline		71.91	50.90	59.61±1.2

Table 2.1: The Performance of sixteen NER Systems for English Language which participate in the CoNLL-2003 Shared Task: Language-Independent NER using a wide variety of machine learning techniques [28].

word are made from information within a sentence: previous and next words of current word, or features based on the orthographic format of a word. e.g., whether a word starts with a capital letter. Global features are extracted from the information of whole document that includes the word, e.g., whether a word has been seen with a title in the same document or not. For instance, if we see the word *"Green"* in a document but it occurs somewhere else in the document with title "Mr." then we can assume that "*Green*" is the name of a person.

In addition to local and global features, some lists are derived from the training data to be used in the feature selection process. For instance, a list of bigrams of the words which precede an entity type because some bigrams like "city of" or "arrives in" can help to detect an entity type. Using a Gazetteer (list of known proper names) also improves results for English.

Curran et al. [8] also used maximum entropy for recognition of named entities in CoNLL-2003. They achieved high results for both English and German language. They have defined some contextual predicates as a baseline system and applied other contextual predicates in their final system. The contextual predicates in baseline system use POS tag and named entity tag for windows of size two.

2.4 Maximum Entropy Model

A good introduction to maximum entropy model can be found in [2]. We introduce the concept of maximum entropy through an example from our anonymization context. Suppose we want to model the probability of a term being anonymizable or not in our corpus. Consider a set of examples **X** and a set of all possible labels **Y**.

$$\mathbf{X} = \{ \text{ set of all terms in the corpus } \}$$
$$\mathbf{Y} = \{ \text{ Anonymizable, NonAnonymizable} \}$$

then the training set **S** is $\{(x,y)|x \in \mathbf{X}, y \in \mathbf{Y}\}$.

The goal is to compute the joint probability distribution $\boldsymbol{p}$ defined over $\mathbf{X} * \mathbf{Y}$. The first step is to extract a set of facts from the samples that will help us construct a model. The first obvious fact or first constraint is:

$$\sum_{x,y} p(x,y) = 1$$

To simplify the example, consider two terms $\{a,b\}$ and the labels "1" as Anonymizable and "-1" as NonAnonymizable. Therefore, the first constraint is

$$p(a,1) + p(b,1) + p(a,-1) + p(b,-1) = 1$$

There are infinite number of models which satisfy this constraint. For Example, if we consider $p(a,1) = 1/2$ and $p(a,-1) = 1/10$, that means the model always choses term "a" as an anonymizable term. The uniform model is another possibility.

$$p(x,y) = \frac{1}{2} \qquad \forall x \in \{a,b\} \quad \textit{and} \quad y \in \{1,-1\}$$

There could be other facts that we might realize from our corpus. For instanse, we may detect that the model chosee either a or b as anonymizable terms in 30% of the times. Then, we integrate this constraint into the model as following.

$$p(a,1) + p(b,1) + p(a,-1) + p(b,-1) = 1$$
$$p(a,1) + p(b,1) = \frac{3}{10}$$

We can discover other facts and include them as other constraints in the model. In the above example, the constraint $p(a,1) + p(b,1) = \frac{3}{10}$ is independent from the context. But we could also consider a constraint dependent on the context surrounding a term. For example, the model chooses the term x as an anonymizable term if **Applicant** follows x. To express this fact we usually use a binary function.

$$f(x,y) = \begin{cases} 1 & \text{if } y = 1 \text{ and nextWord(x) = } \textbf{Applicant} \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

This function is also called a **feature** function and "$nextWord(x) =$ **Applicant**" is an example of a **contextual predicate**. Contextual predicates are some facts that are

determined by the experimenter.

Since a reasonable choice for model p is the uniform model, the problem is how can we find a uniform model subject to a set of constraints.

The maximum entropy method finds a model as uniform as possible, given a set of facts. This model maximizes the entropy $H(p)$ between all the models which satisfy the constraints.

Now, to explain the general mathematical concepts behind the maximum entropy model, consider the random variables $\mathbf{X}$ as a feature vector, including the context of a term, and Y as the set of possible labels. Given a training set

$$S = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_n, y_n)\}$$

we are interested in estimating the conditional probability $p(y|x)$, the probability that a term is anonymizable or not, given the context of the term. The empirical probability distribution $\widetilde{p}$, is defined by

$$\widetilde{p}(x,y) = \frac{\#(x,y)}{n}$$

where n is the total number of samples. When we define some feature function such as equation 2.1, the expected value of function with respect to the empirical distribution $\widetilde{p}(x,y)$ is exactly the statistic we are interested in.

We express the empirical probability of function f s follows.

$$\widetilde{p}(f) = \sum_{x,y} \widetilde{p}(x,y) f(x,y)$$

The expected value of f with respect to the model $p(y|x)$ is

$$p(f) = \sum_{x,y} \widetilde{p}(x) p(y|x) f(x,y)$$

where $\widetilde{p}(x)$ is the empirical distribution of x in the training sample. We constrain this expected value to be the same as the expected value of f in the training sample. We

look for a distribution which is uniform because uniformity means high entropy. The conditional entropy for conditional distribution is

$$H(p) = -\sum_{x,y} \widetilde{p}(x)p(y|x)\log p(y|x)$$

The optimal model is $p^\star = \arg\max_p H(p)$

We seek to maximize $H(p)$ subject to the following constraints:

1. $\sum_{x,y} \widetilde{p}(x)p(y|x)f_i(x,y) = \sum_{x,y} \widetilde{p}(x,y)f_i(x,y) \qquad \forall i : 1..n$
2. $P(y|x) \geq 0 \qquad \forall x,y$
3. $\sum_y p(y|x) = 1$

Using optimization method, the probability distribution which satisfies the above condition is of exponential form.

$$p(y|x) = \frac{1}{z(x)} \prod_{i=1}^{n} \alpha_i^{f_i(x,y)} \tag{2.2}$$

$$\text{where } z(x) = \sum_y \prod_{i=1}^{n} \alpha_i^{f_i(x,y)}$$

The parameters α_i are estimated by an iterative procedure called Generalized Iterative Scaling (GIS) [2]. The number of iterations of GIS, is also a parameter for the model. To implement this mode, we will use the Java-based openNLP maximum entropy package [1]. OpenNLP is an organizational center for open source projects related to natural language processing.

2.5 Conclusion

Named Entity Recognition (NER) is an important first step for many of the natural language processing tasks such as text summarization and question answering. Lexical lookup, rule-based, and statistics & machine learning are main approaches of NER. Despite the fact that rule-based systems obtain better results, it takes too much human

effort for extracting rules, moreover rules are language dependent. Statistics and machine learning have had remarkable success. It reduces human effort and is language independent.

Since our problem is to detect the information related to individuals in judicial decisions for anonymization, we reviewed some tasks done in medical databases such as Datafly and Scrub which are based on lexical lookup and rule-based as well as a work in text documents which is based on machine learning methods.

NOME application aids in anonymization process of judicial decisions based on a simple rule of "first capital letter of a word" (rule-based) and some lists (lexical lookups). Since judicial decisions are long and the number of capitalized words are massive therefore, a long list of capitalized words is highlighted by NOME. Also it takes time to finalize the appropriate anonymizations.

We studied some selected works on NER in the CoNLL-2003 Shared Task. Different learning methods have been applied on the same data and the maximum entropy model is almost always achieved a better performance than other learning methods. This is why, we have selected it as a learning algorithm and we introduced concepts behind this model, binary feature and conceptual predicate which are essential for the implementation.

In the next section, we first introduce how our corpus is built and the words are annotated. Then, we define many features and select some of them using the Sequential Forward Floating Search algorithm (SFFS) for selecting best features and 5-cross validation for calculating the performance of the model. Then, we apply maximum entropy on selected features and test the model on some documents and compare the results with NOME.

CHAPTER 3

ANONYMIZABLE ENTITY FINDER

Nowadays, the Internet is certainly the most widely available information resource. The decisions of the courts, as an important source of information for the lawyers, researchers, media, and public, are available on the Internet. Dissemination and sharing of judicial decisions over the Internet help the public to understand how court decisions are made and ensure people of the openness of justice; however, sometimes this is against the privacy protection. Therefore, certain decisions should be anonymized before publication.

As we have shown in the previous chapters, NOME [15] is used as an assistant application for the anonymization of judgment decisions. This application highlights potential proper names in a document. Using NOME helps reduce the search time for finding anonymizable proper names, however, it cannot detect: 1) the single potential proper names according to its rules (a sequence of two or more capitalized words), 2) the person names which are written in all capitalized letters. In addition, the number of potential proper names proposed by NOME is much more than the number of anonymizable proper names as it was shown in section 1.3.

Since a judicial decision is voluminous and the number of these documents is huge, anonymizing entities is a tedious task. In this work, we try to speed up the process of anonymization by using machine learning algorithms to find anonymizable entities (AEs). As we have shown in table 1.6, finding AEs is more difficult than named entity recognition (NER) because not only we should find the named entities since most of the information of individuals are named entities, but also we should detect which named entities must be anonymized. Coreference resolution of person names and extracting the semantic relationships between named entities are other issues which have to be considered. Coreference resolution of person names is concerned with the detecting of variant forms of proper names that refer to the same person. The semantic relationships between named entities, e.g., relationship between a person and her residence or rela-

tionship between a person with her birthday, also help detect information that leading to the identification of an individual. Therefore, all such relationships must be extracted. However, we did not address these issues in this work.

Our system as an Anonymizable Entity Finder (AEF) tries to find the entities that should be anonymized which is a practical case of information extraction. Since most of the AEs are proper names, we focus on NER which aims to find named entities in texts. As we have shown in the previous chapter, many methods have been applied on NER in Natural Language Processing (NLP) such as Hidden Markov Model (HMM), Support Vector Machine (SVM), and Maximum Entropy (ME). The maximum entropy model as a probability model is applied on several problems in NLP and obtained good results. This is why we have chosen maximum entropy model for our problem.

We consider our problem as a supervised binary classification task. This task classifies objects in the different classes using a training set in which label of classes have already been assigned to objects of the training set. Therefore, we have to annotate each word of a document as an Anonymizable or Non-anonymizable entity. We built a corpus and annotated all words of our corpus.

We have defined several types of features which are presented in section 3.1. Building our corpus is explained in section 3.2. We introduce feature selection methods in section 3.3. Section 3.4 presents cross-validation method for performance estimation of the model and learning algorithm. In section 3.5, we investigate the effectiveness of each defined feature on the model by applying Sequential Forward Floating Search (SFFS) method for selecting a small set of defined features. Section 3.6 shows the results of AEF with the most relevant features.

3.1 Features

Maximum entropy model uses feature functions according to contextual predicates for the calculation of the conditional probability of a class given a context (see section 2.4). The two concepts "*feature*" and "*contextual predicate*" are often used interchangeably. A feature (f) is a binary-valued function and a contextual predicate (cp) is a

portion of the feature (shown in equation 2.1).

The theoretical representation of features is not the same as the one used in the implementation [1]. Basically, features are reduced to the contextual predicates. E.g., equation 2.1 is an example of a binary feature which is reduced to **nextword(context) = "Applicant"** or even to **"next=Applicant"** in the implementation. The maximum number of binary features used is $|cp| \times |T|$, where $|cp|$ is the number of contextual predicates and $|T|$ is the number of possible predicates (tags). Therefore, the number of binary functions is entirely hidden from the user. From now on, in this text, whenever we use term "feature", we mean a "contextual predicate".

All features related to a word in the training data are represented by an event. An event e includes both features of a word and its tag t: $e =< cp_1, cp_2, cp_3, \ldots, cp_n, t >$.

The class (tag) of each word are already determined during the building of the corpus. Features are extracted automatically using the corpus. Each feature in the training data corresponds to a constraint on the model.

For instance, we use two features for our model "current word" and "previous word." The second feature is used only if the current word is capitalized word. For the sentence "Mr. Millar worked at IBM Ltd.", the training data contains six events (see table 3.1).

Event	Current Word	Previous Word	Tag
1	Mr.	-	N
2	Millar	Mr.	A
3	worked	-	N
4	at	-	N
5	IBM	at	A
6	Ltd	IBM	N

Table 3.1: An example of Features and Contextual Predicate of Maximum Entropy Model. Previous word feature is activated if the current word starts with capital letter. For instance, first event is represented by $< currentWord = Mr., N >$ and second event by $< currentWord = Millar, previousWord = Mr., A >$

The first event is represented by $< currentWord = Mr., N >$ and only one binary function (f_1) is activated for this event. The second event is represented by $< currentWord = Millar, previousWord = Mr., A >$ and two binary functions (f_2, f_3) are activated on this

event.

$$f_1(x,y) = \begin{cases} 1 & \text{if } y = N \text{ and currentWord(x) = Mr.} \\ 0 & \text{otherwise} \end{cases}$$

$$f_2(x,y) = \begin{cases} 1 & \text{if } y = A \text{ and currentWord(x) = Millar} \\ 0 & \text{otherwise} \end{cases}$$

$$f_3(x,y) = \begin{cases} 1 & \text{if } y = A \text{ and previousWord(x) = Mr.} \\ 0 & \text{otherwise} \end{cases}$$

We cannot use a numeric feature such as term frequency (tf) in the maximum entropy package[1], but we can define a predicate using a condition based on the term frequency such as "is the term frequency less than a threshold?"

Types of features

We use different types of features. Since the number of occurrences of a feature in each positive and negative class in our data could help find a relevant feature, we first defined some features and after investigations, we decided which features to keep and which new features to add. Then, we checked how a feature improved the performance of the maximum entropy model for detecting AEs and we determined which features are the most relevant.

We classified features into broad classes:

Orthographic Features: They depend on the letters that compose the word. For instance, if a word starts with a capital letter, it could be a proper name or if a word contains a dollar sign, it could be an amount of money. These features are shown in table 3.2.

Dictionary Features: They check if a word appears in some prepared list. For instance, if a word is the name of a weekday, we can consider this word as a Non-Anonymizable entity. Table 3.3 shows these kinds of features. The Common-

Table 3.2: The Orthographic Features used for Maximum Entropy Model

Feature Name	**Token Description & *Motivation***	**Example**
AllLowerCase	All letters are in lower case. *A lower-case word is usually not a proper name.*	made
FirstCap	The first letter is capital. *A proper name in English starts always with a capital letter.*	Robert
InternalCap	Starts with capital letter and contains an internal capital letter. *Some proper names are written in this form.*	MacMinn
AllCaps	All letters are in upper case. *Sometimes a proper name is written with all capital letters.*	LISA
FirstCapEndPeriod	An alphabetical string that starts with a capital letter and ends with a period. *It could be a name, especially when the token is the last word of a sentence and period is not removed from the token. However, titles are written in this form.*	Mr.
AllCapsPeriod	Contains only capital letters and periods. *Judicial documents contain a lot of this kind of strings which are used as an abbreviation. However, they could be proper names.*	D.L.R.
Alphanumeric	Contains at least one digit. *A proper name has not digit.*	F32
LowerCasePeriod	Starts with lower case letter and contains a period. *These tokens are not proper names.*	p.34
OneDigit	Only one digit. *One digit may indicate a date (month or day).*	5
TwoDigits	Made only up of 2 digits. *Sometimes it refers to a date (year or month or day).*	99
Digits	Contains digits with comma or period . *Numbers are not proper names or Anonymized entities.*	30,999
DigitSlash	Made up of digits and slash. *Some dates may are written in this form.*	12/01
Hyphen	Contains a hyphen. *Normally, proper names do not contain hyphens.*	02-fl-502
ContainPunc	Contains a punctuation mark. [1]*This kind of token is not a proper name.*	3:10

[1] punctuation mark = { , ! ? ; ’ : * < = > @ ‘ ^ _ $ % # &~( ) { } [ ] }

Word, MonthName, and WeekDay lists are already prepared. **Anony** is a list of anonymized entities which are dynamically built during the training and the test phases. In training phase, a list of anonymized entities is collected and when an entity is predicted as an AEs during test phase, it is added to the list.

Table 3.3: The Dictionary Features used for Maximum Entropy Model

Name	**Description & *Reason***	**Example**
CommonWord	Token is a common word.[1] *A sentence starts mostly with a common word.*	The
MonthName	Token is a month name. *A month name can help to detect the birth date.*	December
WeekDay	Token is a week day name. *A week day is not an anonymizable entity.*	Friday
Anony	A list of anonymized entities that are collected from the corpus.	

[1] Common word is a commonly used word such as "the" which has a high frequency in a text. A list of these types of words is known as a **stop list** in NLP.

Context Features: The current word (W_0) and the neighbors' words within a window of size ± 2, $(W_{-2}, W_{-1}, W_1, W_2)$ are considered as features. The surrounding words of an anonymizable entity can help detect it.

FirstWord: The first word of a sentence is an ambiguous case. The reason is that always a sentence starts by a word with first letter capitalized and we do not have any clue about capitalization.

Compound Features: This set of features are a mixture of some of the previous features such as orthographic features using the tags of one or two previous words. Table 3.4 presents these features.

We use three prepared lists for generating a compound feature, Related-words list, Title list, and Suffix-organization list. These lists are shown in table 3.5.

Name	Description
C1	W_{-1}="**account**" & W_0 is a number which contains # or -
C2	W_{-1} or $W_{-2} \in$ Related-Words & FirstCap(W_0)
C3	W_1 or $W_2 \in$ Related-Words & FirstCap(W_0)
C4	$W_{-1} \in$ Title & FirstCap(W_0)
C5	W_1="**who**" & FirstCap(W_0)
C6	$W_1 or W_2 \in$ Suffix-Organization & FirstCap(W_0)
C7	Label(W_{-1}) = **A** & FirstCap(W_0)
C8	Label(W_{-2}) = **A** & FirstCap(W_0)
C9	W_{-1} is an anonymized MonthName & W_0 is maximum two digits or it is an ordinal number
C10	(MonthName(W_0) or W_0 is maximum two digits or it is an ordinal number) & (W_{-2} or W_{-1} = "**born**") & previousLabel =**A**

Table 3.4: The Compound Features used for Maximum Entropy Model

Related-Words	father, mother, brother, sister, boy, son, grandmother, grandfather, spouse, wife, husband, daughter, aunt, uncle,cousin, friend, boyfriend, girlfriend, stepmother, stepfather, parent, parties, party, tutor, applicant, respondent, plaintiff, defendant
Title	Mr., Mr, Ms., Ms, Monsieur, Messieurs, MM., MM, Madame, Mesdames, Mme, Mmes, Mrs., Mrs, Mademoiselle, Mesdemoiselles, Mlle, Mlles, Dr., Dr, Me, Miss, Rev., Jr., Sr., Dame, Lord
Suffix-Organization	university, school, hospital, enterprise, ltd, ltd., inc, inc., company, academy, institute

Table 3.5: The Lists used in Compound Features

A word can have more than one feature. For instance, the word "Mr." has two orthographic features *FirstCapEndPeriod* and *FirstCap*. In some cases, a word may have several features which overlap. Two features overlap when a feature can be a subset of another feature, e.g., *FirstCapEndPeriod* is a subset of *FirstCap*. That means that when the feature *FirstCapEndPeriod* is active for a word, then the feature *FirstCap* is certainly active. Therefore, these features are not independent and overlap.

A maximum entropy model can deal with overlapping features. The reason lies in the fact that if all features are independent then the maximum entropy model is the same as maximum likelihood model and the iterative algorithm GIS is not useful [20]. Moreover,

if f_1 and f_2 are two independent features and $f_3 = f_1 \bigcup f_2$, then the result of maximum entropy model using all three features is the same as when any pair of these three features is used. The mathematical reason behind this is explained in Andrew Borthwick's thesis [3] and Ratnaparkhi's thesis [20].

In section 3.5, we will investigate the impact of different combinations of features on the performance of the system and try to find a small set of distinguishing features which can help detect anonymizable entities.

3.2 Building the Corpus

In this section, we discuss the process of building our corpus, the preprocessing of data, and tagging the words in the corpus. We use a collection of 155 judicial decisions (2002), 16 decisions from the Superior Court of Ontario and 139 decisions from the Supreme Court of British Columbia provided by Frédéric Pelletier of LEXUM [1]. For each document we also received the corresponding anonymized version (documents are in MS-Word format).

In our collection, only about 3% of the documents have less than 2 pages and other 97% are long documents. Since finding AEs in long documents is more difficult than short ones, we removed documents that have less than 2 pages (only 4 documents); therefore, our corpus is built based upon 151 documents. After preprocessing of original documents, there are $569,031$ words in our corpus ($13,997$ words are anonymized). The average size of a document is 15 pages, if we consider 250 words per page. Suppose that a document is 15 pages length, finding AEs within such a document is a tedious task.

Since we want to use features such as "*FirstWord*" and "*neighbor words* within a window of size ± 2," we need to know the boundary of a sentence. Sentence boundary disambiguation is the problem of detecting the beginning of a sentence in a text. The punctuation marks '.', '!', and '?', which are mostly placed at the end of a sentence, are usually ambiguous. For instance, when a period is used in an abbreviation form, distinguishing the end (and so the beginning) of the sentence is fuzzy. In the following

[1] http://www.lexum.umontreal.ca/

example, the two words “Mr.” and “B.C.” are the ambiguous cases because the next word for both starts with a capital letter.

He saw Mr. Edgar at B.C. Cancer Agency.

We use a simple rule-based approach for detecting the boundary of a sentence based on two lists; a list of judicial abbreviations which are collected from the law library of the Université de Montréal [2] and a list of person’s titles.

Data preprocessing

For creating the corpus, some preprocessing on the data is required. The first step is to convert the document from Microsoft-Word format to plain text format. We only keep the body of judgment and remove tables from the document. In judicial decisions, each document has a header part which contains information about the parties and the court name. Sometimes, this header is repeated at the end of the document. We also remove these two parts from the document. Tables and the header section that are removed from a document will be considered in a later stage. The information in a header section could help detect anonymizable person names because most anonymized names are the name of the parties.

We use our sentence splitter in order to extract the first word of a sentence. The words are tokenized by white spaces however, some characters are removed from a word, e.g., possessive sign (’s), because finding anonymized entities would not be easy if a word contains additional characters. Also, some strings are removed, such as “[n]” that indicates the beginning of paragraph *n*. The following modifications are performed on a word.

- Removing the whole token when it is a number surrounded by a pair of brackets, such as **[1]** and **[23]**.
- Removing punctuation marks such as {**! ? , : ; ’**} from the end of a word. For example, **said:** converts into **said**.

[2] http://www.bib.umontreal.ca/DR/ressources/abreviations.htm

- Removing a pair of balanced parentheses, brackets, and quotes form a word. For example **(1999)** and **"child"** convert into **1999** and **child**
- Removing any of the symbols {**"** **(** **[**} form the beginning or {**"** **)** **]**} from the end of a word, if the matched symbol is not found in the middle of the word. For example, **fashion)** converts to **fashion** but a word such as **a(ii)** does not change.
- Removing possessive mark. For example, **mother's** converts into **mother**.
- Removing the ellipsis mark from the beginning or end of a word. For example, **family....** converts into **family.**
- Spliting a word which contains **/** or **-** if the all the characters of the word are letters. For example **she/he** converts into two words **she** and **he**.

Finally, we put each word of the document into a separate line of our corpus in order to be able to associate tags to the words in the next step. A period on one single line indicates the sentence boundary (see Table 3.6 for an example).

Tagging the Words

The goal of classification in machine learning is to classify similar objects into similar classes using a training set. The class (tag) of an object in the training set is already determined. Since we defined our problem as a supervised binary classification problem and we hope that AEF can find anonymizable entities, we should determine the anonymized entities in our corpus.

After tokenization, the next step is the tagging of each word of a document in our corpus according to the fact that it is anonymized or not. We consider two classes, Anonymized entities (**A**) and Non-anonymized (**N**) entities. We compare each anonymized document with the original one, line by line, to find anonymized words which we manually tag them as Anonymized Entities in our corpus. Table 3.6 is an excerpt of our Gold Standard corpus.

During this process of tagging anonymized entities of documents, some errors were found in the anonymized version of documents. For instance, some occurrences of a

Word	Tag
The	N
plaintiff	N
and	N
defendant	N
had	N
a	N
relationship	N
and	N
as	N
a	N
result	N
have	N
a	N
child	N
Mika	A
born	N
August	A
14	A
1987	N
.	N
The	N

Table 3.6: An Excerpt of Gold Standard Corpus

proper name were not anonymized, or some entities were anonymized where they should have not been. This indicates the need for automatic anonymized entities finder system for courts.

3.3 Feature Selection

Dimensionality reduction is a challenging problem in machine learning applications. Feature Selection (FS) and Feature Extraction (FE) are two approaches for dealing with dimensionality reduction. Feature selection tries to find a subset of relevant features from the original set of features. Feature extraction tries to generate a new feature by transforming or combining the original features [23]. According to Anil Jain and Douglas Zongker [10], feature selection brings two benefits:

1. Decrease in the number of features that lead to a reduction in the cost of learning.

2. Improvement in the performance of the learning algorithm.

The goal of feature selection algorithms is to detect a subset of the original feature set that increases the performance of learning or without considerably decreasing the performance of the learning algorithm [18]. This problem would need an exhaustive search for finding the optimal subset of features. For instance, there are 2^n different feature subsets where n is the number of original features. In order to find an optimal subset of features, all feature subsets must be evaluated. Therefore feature selection criterion is used to evaluate the efficiency of a feature subset [18]. Higher value of feature selection criterion represents a better feature subset.

There are several feature selection algorithms that are categorized according to search strategies and evaluation criteria [23] [19]. The following taxonomy is based on the search strategies.

Filter methods are independent of a learning algorithm. They rely only on the characteristics of data.

Wrapper methods are based on a learning algorithm and the performance of learning is used as a feature selection criterion. They try to find a subset of features which can improve the performance of learning. Examples of this method are Sequential Forward Selection (SFS), Sequential Backward Selection (SBS), and Sequential Floating Forward Selection (SFFS).

Hybrid is the combination of two previous methods and uses the advantages of both.

"Generally, the wrapper method achieves better performance than the filter method, but tends to be more computationally expensive than the filter approach" [19]. We will use the wrapper method because we want to select a subset of our original features in which the performance of maximum entropy for classifying the Anonymizable and Non-Anonymizable entities can improve.

One of the simplest wrapper methods is the sequential forward (backward) selection method which adds (removes) a feature one by one at each step and evaluates the per-

formance of the learning algorithm until the required number of features is reached [19]. These methods suffer from the problem of "nesting effect."

Nesting effect means that the algorithm cannot remove a feature which was added in previous steps (sequential forward selection) or add a feature which was removed in previous steps (sequential backward selection). Therefore [17] has proposed the sequential forward (backward) floating search (SFFS) method. This algorithm is a combination of forward and backward search. It applies a number of backward steps after each forward step as long as the result of backwarding is better than the last level of forwarding. The algorithm is explained in the following [18].

Algorithm 3.3.1: SFFS(Y, d)

- Input: $Y = \{y_i | i = 1 \ldots n\}$, y_i is a feature and n = number of original features.

- Input: d = number of required features.

- Input: J = evaluation criteria function.

- Output: $X = \{x_i | i = 1 \ldots d\}$

- Select d features among n original features

$X \leftarrow \{\}$

for $i \leftarrow 1$ **to** 2

$$\begin{cases} x^+ = \arg\max_{y_i \in Y - X} J(X \cup y_i) \\ X \leftarrow X \cup x^+ \end{cases}$$

$k \leftarrow 2$

while $(k < D)$

$$\begin{cases} \begin{cases} x^+ = \arg\max_{y_i \in Y - X} J(X + \{y_i\}) \\ X \leftarrow X + \{x^+\} \\ k \leftarrow k + 1 \end{cases} \\ \textbf{do} \\ \begin{cases} x^- = \arg\max_{x_i \in X} J(X - \{x_i\}) \\ \textbf{if } (J(X - \{x^-\} > J(X)) \\ \begin{cases} X \leftarrow X - x^- \\ k \leftarrow k - 1 \end{cases} \end{cases} \\ \textbf{while } (J(X - \{x^-\} > J(X)) \end{cases}$$

3.4 Learning Algorithm

To apply the maximum entropy method, we need a training set to train the model and a test set to evaluate its performance. We use a K-fold cross validation method, as explained in the following, for performance estimation.

K-fold Cross Validation

K-fold cross validation method is applied in machine learning for two purposes.

- Parameter Selection: Machine learning algorithms usually have some parameters for which an optimal combination should be found. For instance, in K-Nearest Neighbors (KNN) algorithm, the parameter that should be optimized is the number of neighbors. The criteria function for parameter selection is the error rate.
- Performance Estimation: When the optimal parameters are chosen for a model, we can use a cross validation method to estimate the performance of the model.

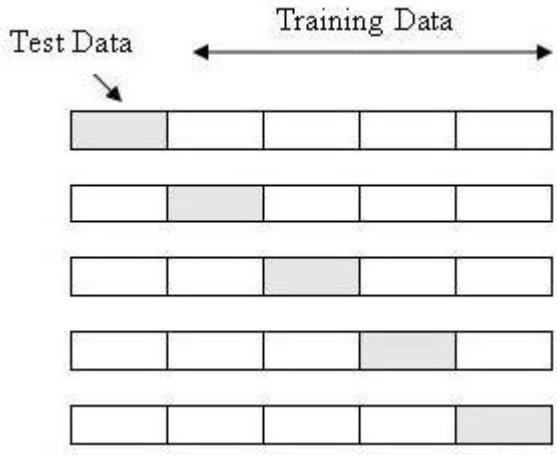

Figure 3.1: The 5-Cross Validation Schema: the experiment applies 5 times using one part an a test set and the rest as a training set.

This method splits the data into K parts and the learning algorithm is applied K times (figure 3.1). At each turn, one part is considered as test set (validation set) and the rest $(k-1)$ parts are treated as training set. The error rate (E) or the performance of the learning algorithm is the average of error rates or performances on k validation sets. In fact, all examples are used for both training and test set.

$$E = \frac{1}{k}\sum_{i}^{k} E_i$$

In our case, we use 5-fold cross validation for performance estimation. The performance of maximum entropy is the average performance of 5 experiments. In each experiment, we select randomly $\frac{1}{5}$ of the documents from our corpus for test set and the rest are considered as training set. Then we generate the candidate features for each word in the documents of training and test set. Some features of the test set have to be generated during the prediction of word tag because they depend on the tags of previous words and we do not know those tags in advance. For instance consider feature $C7$ in table 3.4 which is generated based on the tag of previous word.

Having a training set and a test set which contain the candidate features for words of documents, now we can apply the maximum entropy for creating a model using the training set. Then we predict the tag for each word of the test set using a model data file. Finally, we calculate the precision, recall, F_1, and F_2-measures for performance evaluation. Figure 3.2 shows this process.

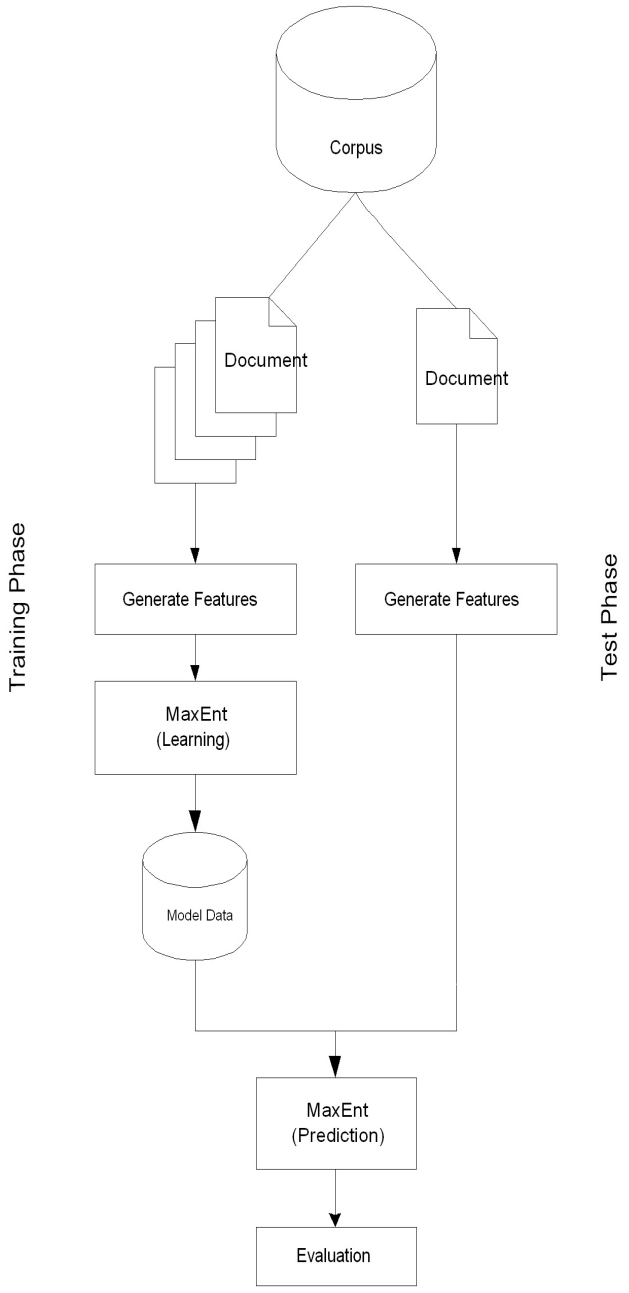

Figure 3.2: The Structure of Learning Algorithm for each Experiment in Cross-Validation

3.5 Experiments

We have done our experiments with openNLP Maximum entropy Java package [1]. OpenNLP [3]is an open source natural language processing library. In the maximum entropy package [1], there are two parameters that should be set by the user: the number of iterations for Generalized Iterative Scaling (GIS) and a number called *cutoff* that shows the minimum number of times that a feature must have been seen during training.

For determining the number of iterations of GIS, we applied a 5-fold cross validation using all defined features. We applied a maximum entropy model for 10, 20, 30, 40, 50, 100, 150, ..., 550 iterations for each set and selecting zero for *cutoff*. Figure 3.3 shows the changes of error rate in terms of the number of iterations.

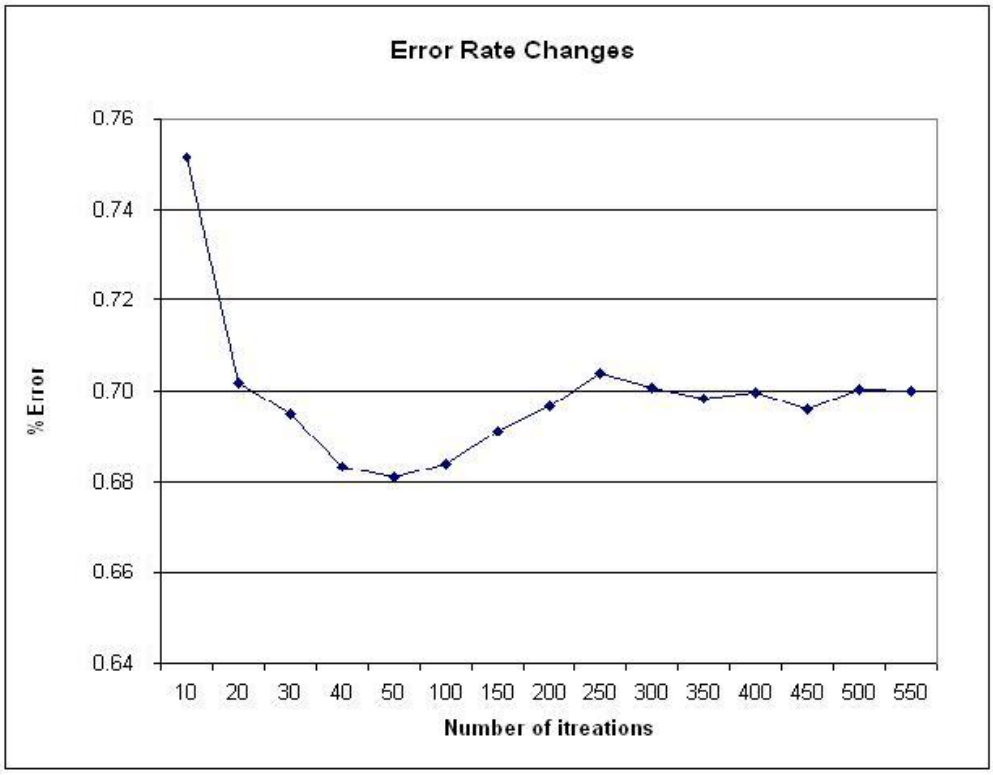

Figure 3.3: The Changes of Error Rate in terms of Number of Iterations for Maximum Entropy Model using all Defined Features

[3]http://opennlp.sourceforge.net/

At each iteration, GIS algorithm estimates new values for the parameters of the model (α_i) which fit the constrains better than its predecessors. Therefore, the error rate decreases but when the algorithm starts to overfit the data, the error rate rises. The algorithm should be stopped when the error rate starts to increase at certain number of iteration (early stopping).

Since the error rate is going up after 50 iterations therefore, we consider 50 as the number of iterations and then we verify how the defined features affect the performance of the model. The reason for selecting zero for *cutoff* is that a word needs at least one feature but by setting *cutoff* greater than zero it is possible that some words are ignored due to lack of sufficient number of features depending on the selected feature set.

The number of iterations depends on the degree of overlapping of features and it is approximately proportional to the number of active features for each context [3]. We built different anonymizable entity finders using maximum entropy model with different feature collections. We fixed two parameters of the maximum entropy package [1], the number of iterations is set to 50 and cutoff is set to 0 for all further experiments. The sequential forward floating search method is applied to select a small set of features that are most relevant.

Features Selection and Analysis

We apply maximum entropy model (using package [1]) on training and test sets with different candidate feature set using 5-cross validation. The performance of the system is measured by calculating the percentages of precision, recall, and F-measure on test sets in each experiment. Since we want to find all AEs and that a high recall indicates that most of AEs are found, we use F_2-measure (see section 1.4 as a criteria function for SFFS because the F_2-measure is used when recall is more important than the precision.

All features (Orthographic features, Dictionary features, Compound features, First-Word feature, Context features) are used. The context features are different from other features. For instance, as we have shown in section 3.1, previous word feature W_{-1} is a contextual predicate and maximum number of binary functions which maximum entropy uses for the contextual predicate W_{-1} are $n \times |T|$, where n is the number of words and

$|T|$ is the number of possible tags.

For the neighbors' words within a window of size ± 2 of current word, we explored different approaches of using these types of features. We evaluated the performance of maximum entropy for each one of the context features W_{-1}, W_{-2}, W_1, and W_2 individually as follows.

1. A context feature is used for all words in a document. In this case the performance is low for all context features.

2. A context feature is used for words that their first letter is not capitalized. In this case, the performance of maximum entropy is low except for W_{-1} feature.

3. A context feature is used only for capitalized words. Since proper names start with capital letters therefore using previous word e.g., W_{-1} can help detect the proper names. For instance, title words such as "Mr." or "Ms." are always the previous word of persons' names. The performances of maximum entropy for all context features is better than two previous cases.

Therefore, the context features (i.e., W_{-1}, W_{-2}, W_1, W_2) are used only for capitalized words. Also, an another feature SW_{-1} is used. SW_{-1} is the first previous word of current word when the current word does not start with capital letter.

Applying Sequential Forward Floating Search

We applied the SFFS method to select a small set of features. First, SFFS evaluates the performance of all individual defined features and selects a feature that has a maximum F_2-measure. The first previous words (W_{-1} and SW_{-1}) are significant features for the model (see figure 3.4).

Feature W_{-1} has the best performance as shown in table 3.7. Feature $C4$ that can detect the persons' names using a list of title is also a good single feature for the model. The performance of the system with adding features *FirstCap* and *Anony* to candidate

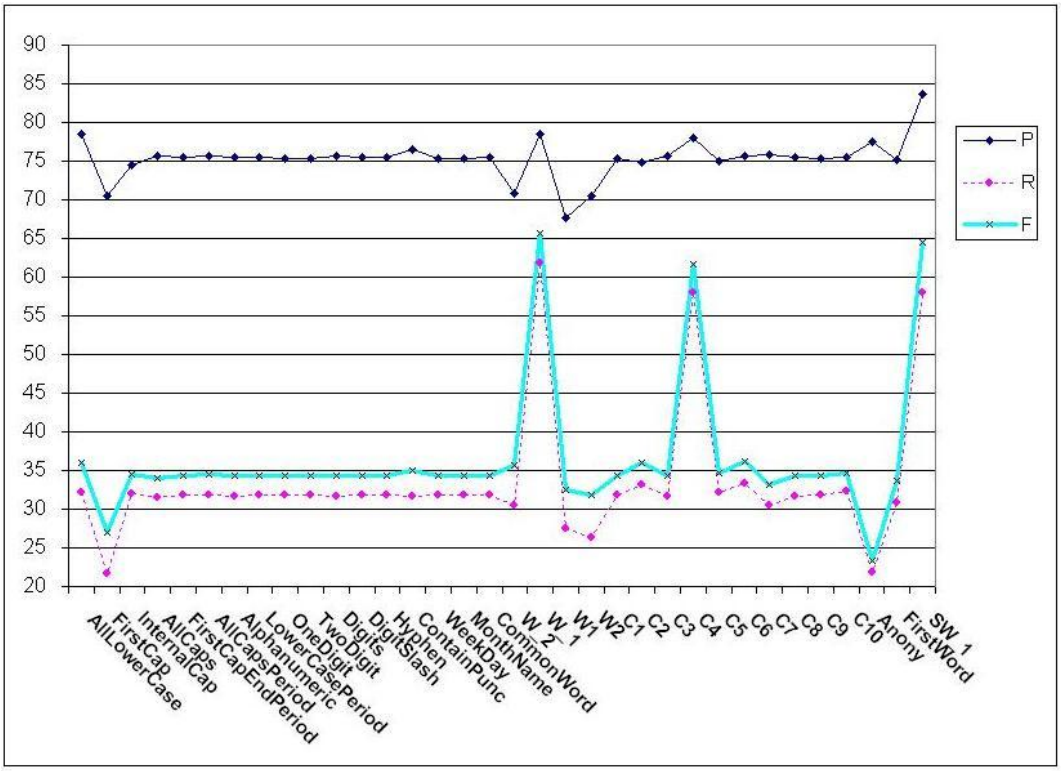

Figure 3.4: The Changes in Precision, Recall and F_2-measure for all Individual defined Features. Features W_{-1}, W_{-2}, W_1, and W_2 are used for capitalized words. The feature SW_{-1} is the previous word of current word which does not start with capital letter.

feature set, is always low in all experiments. Feature *FirstCap* cannot improve the performance of the model unless it is used as a condition in other features, e.g., context features. The performance of the system with other features is almost the same.

SFFS adds W_{-1} to the feature collection due to its high F_2-measure, and tries to find another feature which can improve the performance of maximum entropy model. The next selected feature is SW_{-1} (see figure 3.5).

The third selected feature (figure 3.6) is *AllCapsPeriod*. The performance of Maximum entropy model with each new selected feature is shown in table 3.7. A row of this table shows the performance of feature collection which contains the previous features (previous rows) and new selected feature (current row). In this level, the F_2-measure of the features W_{-2} and $C9$ are high and SFFS will be selected them in next steps.

After selecting 3 forwardings, SFFS starts backwarding and removes features one

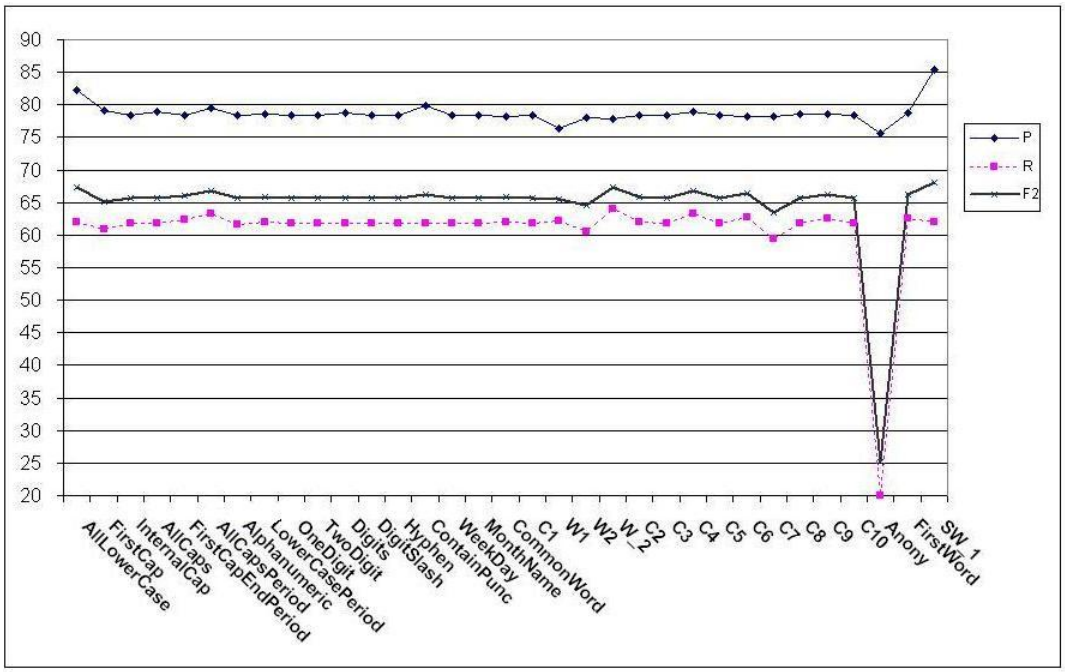

Figure 3.5: The Changes in Precision, Recall and F_2-measure using W_{-1} Feature with the rest of Features

by one from our last feature collection that has high performance, to evaluate whether removing a feature can improve the performance of model or not. If it cannot improve the performance then SFFS continues to add a new feature to our collection. In our experiments, backwarding did not help. That means, always the result of last forwarding was higher than the best result of backwarding.

The fourth selected feature is $C9$ that can help detect a birth date that should be anonymized. Table 3.7 shows the performance of selected feature with the previous set in each forwarding step. As this table shows, after adding ninth feature the performance of maximum entropy does not change significantly.

our experiments are shown that the context features are the most important features. In fact some compound feature are embedded in these features. For instance, $C4$ feature is activate if previous word is a title and current word starts with capital letter. Since we consider previous word of a capitalized word (W_1 feature) then for each title that used in our corpus, we have one contextual predicate. For example, W_1=Mr.

Set	Features	Precision	Recall	F_1-measure	F_2-measure
1	W_{-1}	78.43	61.82	68.16	65.69
2	SW_{-1}	85.33	61.91	71.66	68.07
3	AllCapsPeriod	86.19	63.61	73.05	69.58
4	$C9$	86.29	64.28	73.53	70.14
5	W_{-2}	85.34	65.13	73.82	70.67
6	FirstCapEndPeriod	84.78	68.27	75.59	72.98
7	AllCaps	86.09	71.16	77.86	75.48
8	CommonWord	85.92	72.07	78.34	76.12
9	W_1	86.57	74.59	80.11	78.18
10	C6	86.50	74.88	80.25	78.37
11	ContainPunc	86.67	74.92	80.35	78.45
12	MonthName	86.67	75.03	80.41	78.53
13	C2	86.65	75.11	80.45	78.58
14	TwoDigits	86.66	75.13	80.47	78.60
15	Hyphen	86.69	75.16	80.50	78.63
16	WeekDay	86.67	75.20	80.51	78.66

Table 3.7: The Changes of the Performance of Maximum Entropy Model with Selected Feature Set

We cited in the end of section 3.1, if f_1 and f_2 are two independent features and f_3 = $f_1 \bigcup f_2$, then the result of maximum entropy model using all three features is the same as when any pair of these three features is used. The features C2, C3, C4, C5, and C6 depend on context features. For instance, the binary function f_1 correspond to contextual predicate W_1=Mr. and binary function f_2 correspond to contextual predicate W_1=Ms. are a subset of binary function f_3 correspond to contextual predicate C4 therefore we can remove C4 feature if all titles are represented in our corpus. As figure 3.4 shows the C4 feature is a good single feature but during feature selection it could not improve the performance of the system.

We selected a collection of the first 9 features listed in table 3.7 { W_{-1}, SW_{-1}, *AllCapsPeriod*, $C9$, W_{-2}, *FirstCapEndPeriod*, *AllCaps*, CommonWord, W_1 } as a baseline features collection. The performance of Anonymizable Entity Finder (AEF) is shown in table 3.8.

In a document, an entity may have many occurrences. If one of them tagged as an anonymizable entity (AE) then all such entities should also be tagged as AE because

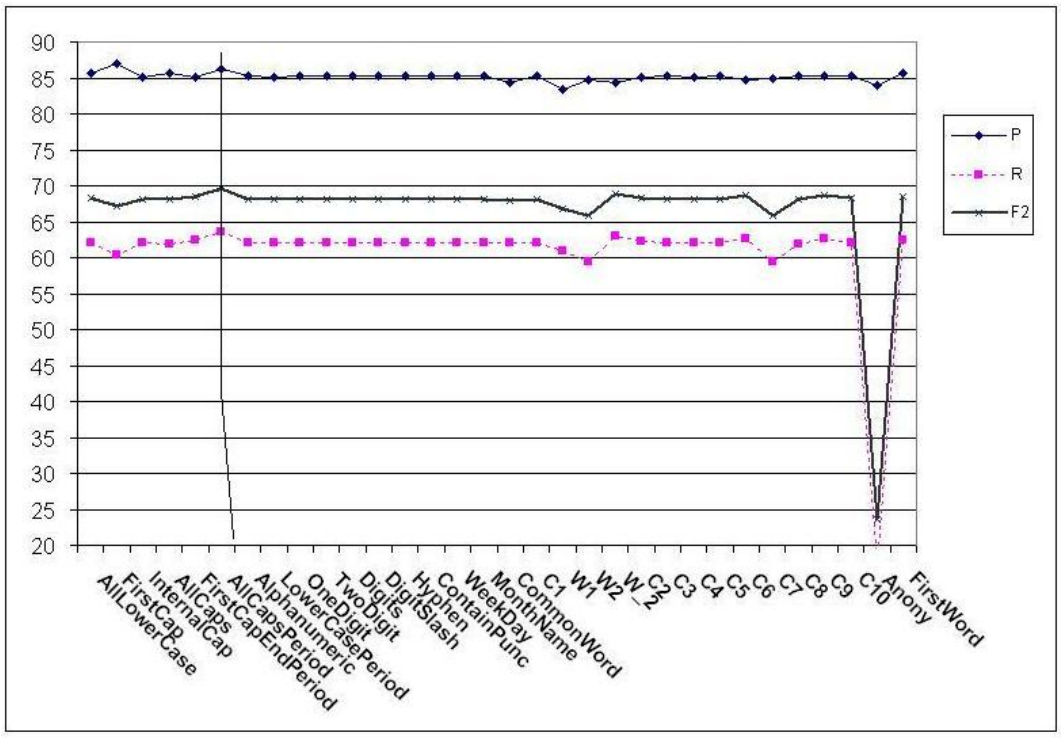

Figure 3.6: The Changes in Precision, Recall and F_2-measure using W_{-1} and SW_{-1} Feature with all other Individual Features

Precision	Recall	F_1	F_2
86.57	74.59	80.11	78.18

Table 3.8: The Performance of Anonymizable Entity Finder (AEF)

when an entity must be anonymized in a document then all occurrences of such entity should be anonymized.

Since prediction of word's tag depends on the local context of word, consequently the maximum entropy may predict different tags for a word in same document. This problem is known as coreference resolution which we did not address it in our work. As we encountered with this problem that a person's name labeled as AE in one place and as Non-AE elsewhere in the same document during testing the system.

Accordingly we think the performance of Anonymizable Entity Finder is reasonable. To the best of our knowledge there is no similar system for finding anonymizable entities using machine learning techniques in the justice domain such that we are able to compare

the performance of AEF with their performance. The only system that currently assists in the processing of anonymization in justice domain is NOME that tries to find potential proper names, but not anonymizable entities. Therefore, comparing these two system is difficult.

The result of NOME is a list of potential proper names (terms) which usually contains much noise, noise being a term that is not proper name e.g., "Income Taxes".

The result of Anonymizable Entity Finder is tags for words (AE or Non-AE). Since the anonymizable entities are important for us, and we are able to compare AEF with NOME, the output of AEF is represented with a list of anonymizable entities. Therefore if the system labels a word as an anonymizable once, we show it in the list.

We compared the our result with the result of NOME on same documents according to the criteria shown in the next section.

3.6 Evaluation

After choosing the baseline feature collection, we must apply the learning algorithm on the corpus to create a maximum entropy model. This model gives us a model data file that AEF uses to detect the anonymizable entities of a document. For finding the anonymizable entities of a new document, we convert the MS-Word document into a text format and remove the head section. Then we apply the maximum entropy modelfor predicting the tag of each word. The output of AEF is a list of words that are labeled as anonymizable entities. Figure 3.7 illustrates the process of finding anonymizable entities of a new document using AEF.

To evaluate AEF and compare it with NOME, we selected randomly 149 documents for training set and 2 documents for evaluating the model. We applied maximum entropy on the training set with a baseline feature collection. Then we tested the model on 2 documents.

Since we are not dealing with the problem of coreference resolution and that finding an AE is important for us, our result is a list of entities that are tagged as anonymizable even if it was tagged as anonymizable only once in the document. As a result, if a word is

predicted once as an anonymizable entity we consider it as anonymizable in our results.

For evaluation documents, we selected a document from the Ontario Superior Court of Justice (Appendix I, henceforth Document I) and a document from the Supreme court of British Columbia (Appendix II, henceforth Document II). We removed the head sections of these files and applied both NOME and AEF.

To be able to compare these two systems, we considered two criteria:

1. How many distinct Anonymizable Entities (AEs) are recognized by each system and what is the total number of occurrences ?

2. How many distinct Non-anonymizable Entities (Non-AEs) are in the results of each system and what is the total number of occurrences ?

Evaluation of Document I

There are 9 distinct proper names and 3 birth dates that must be anonymized in Document I. Table 3.9 shows the results of two systems. NOME found only 7 persons' names. Three birth dates are never shown by NOME. However, we can find all birth dates in a document by adding the word "born" to the inclusion list of NOME.

AEF detected about all distinct proper names except one that has only one occurrence. Moreover, AEF found 2 birth dates that should be anonymized. The performance of AEF for Document I is shown in table 3.11.

All occurrences of AEs cannot be detected by AEF because our system labels a word according its local context. However, if an AE is detected by NOME, then all its occurrences are simply listed in the result.

	Proper Names		Date		
	Distinct	All Occ.	Month	day	Total
Gold	9	449	3	3	455
NOME	7	388	-	-	388
AEF	8	336	2	2	340

Table 3.9: The Number of Anonymized Entities that are Recognized by both NOME and AEF in the Document I of the Gold Standard Corpus

As table 3.10 shows, both results have some noise. That means, there are some words in both results that are not AEs. The total number of distinct words that are not AEs in the result of NOME is much more than AEF results.

	Non-Anonymizable Entities	
	Distinct	All Occ.
NOME	51	119
AEF	7	14

Table 3.10: The Number of Non-anonymized Entities in the Results of both NOME and AEF for Document I

Precision	Recall	F_1	F_2
96.05	74.73	84.05	80.70

Table 3.11: The Performance of AEF for Document I

Evaluation of Document II

There are 8 distinct proper names and 3 birth dates that must be anonymized in Document II. As table 3.12 shows, AEF found all distinct proper names while NOME found only 5 of them. In addition, only one day of birth cannot be detected by AEF.

Both results contain many words that are not AEs. There are many person names with a title in this document that are not anonymized, consequently the number of Non-AEs is large for both results. However, the number of distinct Non-AEs in the result of NOME is about quadruple the AEF (see table 3.13).

	Proper Names		Date		
	Distinct	All Occ.	Month	day	Total
Gold	8	86	3	3	92
NOME	5	26	-	-	26
AEF	8	70	3	2	75

Table 3.12: The Number of Anonymized Entities that Recognized by both NOME and AEF in the Document II of the Gold Standard Corpus

	Non-Anonymizable Entities	
	Distinct	All Occ.
NOME	102	253
AEF	27	62

Table 3.13: The Number of Non-anonymized Entities in the Results of both NOME and AEF for Document II

The performance of AEF is shown in table 3.14. Precision of AEF for this document is low because there are many entities that labeled as AEs. However recall is high because most of AEs are detected by system.

Precision	Recall	F_1	F_2
54.74	81.52	65.50	70.09

Table 3.14: The Performance of AEF for Document II

3.7 Conclusion

Judicial decisions must be anonymized before publication. The anonymization task is tedious because a judicial decision is voluminous and the number of these documents is huge. We used a machine learning method to find anonymizable entities in judicial decisions. The problem is treated as a supervised binary classification that classifies the entities into two classes: Anonymizeble and Non-anonymizable. For the classification algorithm, we used maximum entropy model.

In section 3.1, we explained the concept "*feature*" that maximum entropy model uses and how the features are represented in the implementation. Several types of features are used: Orthographic features (table 3.2), dictionary features (table 3.3), context features (the word itself and its neighbor word), *FirstWord* feature, and compound features (table 3.4). The context features W_{-1}, W_{-2}, W_1, and W_2 are used only for capitalized words. Also, another feature SW_{-1} is the previous word of current word when the current word does not start with capital letter.

A word can have more than one feature that are either independent or dependent (overlap). The maximum entropy model works well with overlapping features.

We built a corpus using a collection of 151 judicial decisions because a classification algorithm needs an annotated training data. A sentence boundary detection is needed for using some features, e.g., context features and *FirstWord*. For detecting the boundary of sentences, we used a simple rule-based approach based on two lists; a list of juridical abbreviations and a list of person's titles. To polish the words of corpus we did some preprocessing on the data, e.g., removing "'s".

The words of corpus are tagged as an Anonymied Entities(AEs) or Non-Anonymized Entities (Non-AEs) by comparing the anonymized version of a document with the original one. During finding AEs by comparing the original document with its anonymized version, we discovered that there are still entities in the anonymized version that must be anonymized or some entities were anonymized where they should have not been.

In section 3.5, we tried to find a small set of distinguishing features which can help detect anonymizable entities by applying a feature selection method. Feature selection methods can improve the performance of learning algorithm by finding a small subset of relevant features from the original set. However, an exhaustive search is needed for finding the optimal subset of features.

There are three feature selection methods based on the search strategies: filter, wrapper, and hybrid. Since we want to select a subset of our original features in which the performance of maximum entropy can improve, we used the wrapper method Sequential Forward (backward) Floating Search (SFFS) (see algorithm 3.3.1).

We applied the maximum entropy model using 5-fold cross validation for performance estimation of the model. Precision, recall, F_1, and F_2-measures are utilized for evaluation of performance.

The openNLP Maximum entropy Java package is used for doing our experiments. The number of iterations for GIS is set to 50 and *cutoff* is set to zero for all experiments.

We applied SFFS method for selecting a small set of defined features. F_2-measure is used as a criteria function for SFFS. SFFS starts to evaluates the performance of all individual defined features and selects a feature that has a maximum F_2-measure. The first previous words (W_{-1} and SW_{-1}) are significant features for the model (figure 3.4).

In our experiments, backwarding of SFFS did not help to the feature selection pro-

cess. Table 3.7 shows the performance of selected feature in each forwarding step of SFFS. As this table shows, after adding ninth feature the performance of maximum entropy does not change significantly. Therefore, we chose the first nine features as a baseline feature set. The performance of baseline features set is shown in table 3.8.

We encountered a problem that a person's name is labeled as AE in one place and as Non-AE elsewhere in the same document during testing the system. The reason is that predicting the tag of word depends on the local context of the word, consequently, the maximum entropy predicts different tags for a word in the same document. This problem is known as *coreference resolution* which we did not address in our work.

To the best of our knowledge, there is no similar system for finding anonymizable entities using machine learning technique in justice domain such that we are able to compare the performance of AEF with their performance. The only system that assists in the processing of anonymization in justice domain is NOME.

NOME highlights potential proper names in a document base on finding a sequence of two or more capitalized word and using some list (lexical lookup). However, it cannot detect a single proper name, and the proper names which are written in all capitalized letters. In addition, the number of potential proper names proposed by NOME is much more than the number of anonymizable proper names.

The result of Anonymizable Entity Finder is tag of words (AE or Non-AE). AEF highlights anonymizable entities in a document using machine learning method. Since NOME tries to find potential proper names not anonymizable entities, therefore, comparing NOME and AEF is difficult.

to evaluate AEF, we compared the our result with the result of NOME on same documents according to two criteria: the number of distinct AEs are recognized by each system and the number of distinct Non-AEs that are in the results of each system.

AEF could find about all proper names that should be anonymized but NOME detected 70% of them in average. Moreover The number of distinct Non-Anonymizable entities in result of NOME is about 4.5 times more than those of AEF.

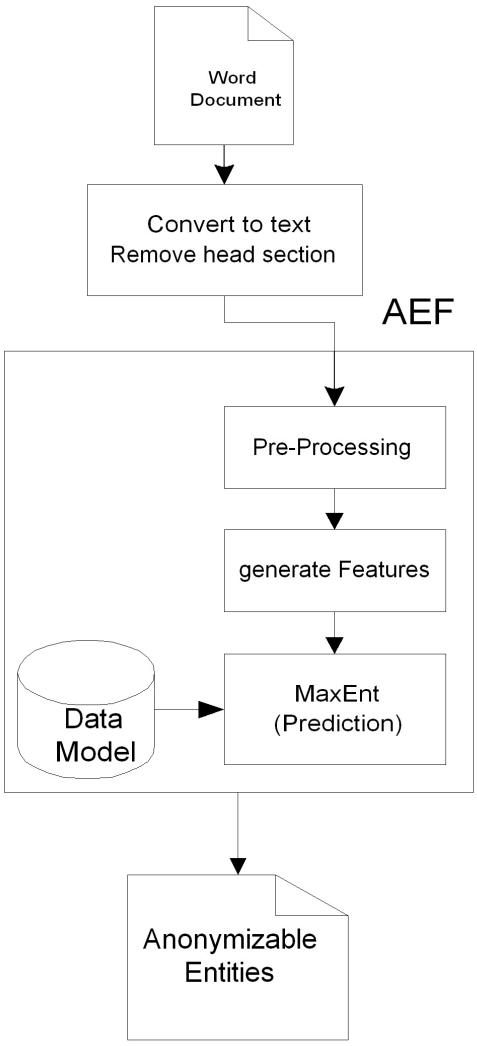

Figure 3.7: The Process of Finding Anonymizable Entities of a New Document using AEF

CHAPTER 4

CONCLUSION

Anonymizing personal information in judicial documents involves two steps. First, detecting personal information that should be anonymized within a document; second, removing, replacing, or concealing this information. The first step is a challenging task in the domain of Information Extraction and Natural Language Processing (NLP).

In this thesis, we proposed a solution named Anonymizable Entity Finder (AEF) for the first step. AEF used a supervised machine learning approach for classifying the entities of a document into two classes: Anonymizable and Non-anonymizable. We selected maximum entropy model as the learning method, because it has achieved a better performance than other learning methods for several NLP tasks. Various types of features are used, among which context features have a great influence on performance. We estimated the performance of AEF using a 5-fold cross validation on 151 judicial decisions and we obtained good results.

To the best of our knowledge, there is no similar system for finding Anonymizable Entities (AEs) using machine learning techniques in the justice domain. Therefore, we were unable to compare the performance of our system with other competitors. The only system that is close to our work is NOME. This system is used as an assistant tool in the process of anonymization in the justice domain. NOME finds only the potential proper names but not the AEs. Moreover, NOME cannot find some of the proper names (e.g., a first name alone) that must be anonymized; furthermore, the number of suggested proper names is much more than the number of anonymizable proper names.

To evaluate our system, we compared the result of AEF and NOME on the same documents based on two criteria: the number of distinct AEs that are recognized by each system, and the number of distinct Non-AEs that are in the results of each system. The results showed that AEF finds all proper names that should be anonymized, but NOME detects 70% of them on average. Moreover, the number of distinct Non-AEs in the result of AEF is 4.5 times less than those of NOME. Our work raised two important points:

- To be able to find AEs in a document, learning algorithms need to learn from the training data. Therefore, having sufficient data is important to learn the machine well. For instance, if we want to extract the account number of individuals, we need to have such information in the training data.

- In a classification algorithm, it is important that data is annotated correctly. We observed some errors in the anonymized version of documents, therefore supervision of an expert is required in the course of annotating the corpus.

4.1 Future Work

There are at least two problems that we did not address in our work. First, an entity may have many occurrences with several references in a document. For instance, an individual may be referred to by last name in one place, but by a nickname in other places in a document. Then all names (first name, last name, or nickname) that refer to the same person must be anonymized and replaced with the same string. This problem is known as coreference resolution. To solve this problem, it would be better to choose a model that can predict labels based on the global information of a word. This is doable by combining features from all occurrences since each occurrence might contain different useful information. A model that is able to cope with this problem is a Markov Random Field model which is based on Maximum Entropy model and is known as Conditional Random Field (CRF) [9].

Second, it would be interesting to take into account the semantic relationships among entities that should be anonymized (e.g., the relationship between a person and her residence or her birthday). Adding this feature would enable the system to detect different anonymizable entities and improve the recall.

We considered only the body of the judicial decision and removed the head section. The head section of a judicial decision contains the name of parties (e.g., defendant and plaintiff) which are generally anonymized. It would be useful to take into account the information of the head section for finding AEs in the body section.

In this study we considered a binary classification method. Since there are several

types of information about an individual that should be anonymized, e.g., birth date and address, it would be interesting to study whether a multi-class classification would improve the performance or not.

It would be interesting to apply other systems for tagging, e.g., Part Of Speech (POS) tagger system or NER system ANNIE, then feed these tags into AEF in the "features collection step."

The judicial decisions that we used in our study are family matter. We would like to apply our system to other kinds of judicial decisions such as criminal matter.

We think machine learning methods outperform rule-based by incorporating contextual information learned from a massive corpus of data. It would be instructive to apply other machine learning algorithms such as Hidden Markov Model, Support Vector Machine, and Conditional Random Field as supervised machine learning methods. Methods provide a higher recall are the best for finding anonymizable entities within data.

In our work, most of the anonymizable entities is persons' names, organization names, and birth dates. In order to detect other information about an individual, such as those mentioned in table 1.3, using the same techniques we have to annotate more data. Since annotation of data takes much time, another alternative is to apply unsupervised learning techniques.

Integrating AEF within the word processor (currently MS-Word), such that an editor can easily edit anonymizable entities in a document, would also be a fruitful endeavor.

Although our experiments were very promising, there is still much more work to do.

BIBLIOGRAPHY

[1] Jason Baldridge, Tom Morton, Gann Bierner, and Eric Friedman. Maximum entropy model version 2.4.0, http://maxent.sourceforge.net/, 2005.

[2] Adam L. Berger, Stephen Della Pietra, and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22 (1):39–71, 1996.

[3] Andrew Borthwick. *A Maximum Entropy Approach To Named Entity Recognation*. PhD thesis, New York University, 1999.

[4] Michael Chau, Jennifer J. Xu, and Hsinchun Chen. Extracting meaningful entities from police narrative reports. In *The National Conference for Digital Government Research*, pages 271–275, Los Angeles, California, 2002.

[5] Hai Leong Chieu and Hwee Tou Ng. Named entity recognition: A maximum entropy approach using global information. In *The 19th International Conference on Computational Linguistics (COLING 2002)*, pages 190–196, Taipei, Taiwan, 2002.

[6] Hai Leong Chieu and Hwee Tou Ng. Named entity recognition with a maximum entropy approach. In *The Seventh Conference on Natural Language Learning at HLT-NAACL*, pages 160–163, Edmonton, Canada, 2003.

[7] Canadian Judicial Council. Use of personal information in judgments and recommended protocol, 2005.

[8] James Curran and Stephen Clark. Language independent ner using a maximum entropy tagger. In *the Seventh Conference on Natural Language Learning (CoNLL-03)*, pages 164–167, Edmonton, Canada, 2003.

[9] Lise Getoor and Ben Taskar, editors. *Introduction to Statistical Relational Learning*. The MIT Press, 2007.

[10] Anil K. Jain and Douglas E. Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE Transactions Pattern Analysis and Machine Intelligence.*, 19(2):153–158, 1997.

[11] Davis Jesse and Goadrich Mark. The relationship between precision-recall and roc curves. In *the 23rd international conference on Machine learning*.

[12] Marie-Francine Moens. *Information Extraction: Algorithms and Prospects in a Retrieval Context*. Springer, 2006.

[13] Department of Justice. Canada's Court System, http://www.justice.gc.ca/eng/dept-min/pub/ccs-ajc/, Avril 2008.

[14] Frédéric Pelletier. Decisions redaction guidelines. Technical report, 2006.

[15] Luc Plamondon, Guy Lapalme, and Frédéric Pelletier. L'assistant d'anonymisation NOME. In *Journées Internet pour le Droit*, Paris, 2004.

[16] Luc Plamondon, Guy Lapalme, and Frédéric Pelletier. Anonymisation de décisions de justice. In Dans Bernard Bel and Isabelle Martin, editors, *XIe Conférence sur le Traitement Automatique des Langues Naturelles (TALN 2004)*, pages 367–376, Fés, Maroc, 2004.

[17] P. Pudil, F.J. Ferri, J. Novovičová, and J. Kittler. Floating search methods for feature selection with nonmonotonic criterion functions. In *Proceeding of the 12th International Conference on Pattern Recognition*, volume B 279-283, Jerusalem, Izrael, Los Alamitos, 1994. IEEE Computer Society Press.

[18] P. Pudil, J. Novovičová, and Kittler J. Floating search methods in feature-selection 15: (11). *Pattern Recognation Letters (ELSEVIER)*, 15:1119–1125, 1994.

[19] P. Pudil, J. Novovičová, and P. Somol. Notes on the evolution of feature selection methodology. *The Journal of the Czech Society for Cybernetics and Information Sciences (Kybernetika)*, 43:713–730, 2007.

[20] Adwait Ratnaparkhi. *Maximum Entropy Models for Natural Language Ambiguity Resolution.* PhD thesis, University of Pennsylvania, 1998.

[21] Justice Department of Nova Scotia. Justice Department of Nova Scotia, http://www.gov.ns.ca/just/divisions/im/foipop/privacy.asp, Summer 2007.

[22] Thamar Solorio. *Improvement of Named Entity Tagging by Machine Learning*. PhD thesis, The University of Texas, 2004.

[23] P. Somol, J. Novovičová, and P. Pudil. Are better feature selection methods realy better? In *Proceedings of the International Joint Conference on Biomedical Engineering Systems and Technologies - BIOSTEC 2008*, pages 246–253. INSTICC Press, 2008.

[24] Latanya Sweeney. *Computational Disclosure Control, A Primer on Data Privacy Protection*. PhD thesis, MIT, 2001.

[25] Latanya Sweeney. k-anonymity: A model for protcting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems,*, 10(5):557–570, 2002.

[26] Gyorgy Szarvas, Richard Farkas, Szilard Ivan, Andras Kocsor, and Robert Busa-Fekete. An iterative method for the de-identification of structured medical text. In *The i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data at AMIA*, Washington, DC, 2006.

[27] Ricky k. Taira, Alex A. T. Bui, and Hooshang Kangarloo. Identification of patient name references within medical documents using semantic selectional restrictions. In *AMIA , Annual Symposium*, pages 757–761, Los Angeles, CA, 2002.

[28] Erik F. Tjong Kim Sang and Fien Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In Walter Daelemans and Miles Osborne, editors, *The Conference on Natural Language Learning (CoNLL-2003)*, pages 142–147, Edmonton, Canada, 2003.

[29] Wikipedia. Privacy, http://en.wikipedia.org/wiki/privacy, 2007.

Appendix I

File: 98-Fl-25133.doc

Due to privacy issues, the original documents are not copied here. Please refer to the digital file: 98-Fl-25133.DOC

Appendix II

File: 2002BCSC1618.doc

Due to privacy issues, the original documents are not copied here. Please refer to the digital file: 2002BCSC1618.DOC