Recent Advances in Intelligent Information Systems
ISBN 978-83-60434-59-8, pages 77–89

Named Entity Recognition in Machine Anonymization

Filip Graliński[1], Krzysztof Jassem[1], Michał Marcińczuk[2], and Paweł Wawrzyniak[1]

[1] Adam Mickiewicz University, Faculty of Mathematics and Computer Science, Poznań, Poland
[2] Wrocław University of Technology, Faculty of Computer Science and Management, Wrocław, Poland

Abstract

The paper presents a formalism for the rule-based Named Entity Recognition (NER). In comparison to existing solutions the new features of the formalism are: applicability for inflected languages and translation of Named Entities. The solution has been implemented for two purposes: Machine Anonymization and Machine Translation. Both applications need their specific grammars but use the same parsing mechanism that reads the rules and recognizes Named Entities in given texts. For the anonymization purposes the application assigns one of predetermined types to each recognized entity. For the purposes of Machine Translation the recognized entities are assigned their types (consistent with a semantic ontology used by the translation system), as well as their equivalents in the target language. Here, we focus on Machine Anonymization. Tools for semi-automatic correction of results of Machine Anonymization are also presented.

Keywords: named entity recognition, machine anonymization, automatic de-identification

1 Introduction

The practical goal of our studies has been to develop a mechanism that will help correctly process Named Entities (NEs) in Machine Anonymization and Machine Translation systems.

The need for this research has been created by the project "A Machine Translation system developed for improvement of public security"[1] undertaken at Adam Mickiewicz University, Poznań under the auspices of Polish Platform For Homeland Security. The aim of the project is to tune existing Machine Translation algorithms and resources for domain-specific translation. One of the first and crucial tasks in the project has been the collection of domain specific resources, namely: lexicons and texts (monolingual and bilingual) characteristic of public security (mainly police texts). However, most texts from this domain contain sensitive information.

[1] Grant No 003/R/T00/2008/05 financed by the Polish Ministry of Science and Higher Education

Such texts may be revealed only after the process of anonymization, which consists in deleting (or replacing) several types of Named Entities (NEs). The same problem was encountered in the project "Technologies for processing and distributing verbal information in internal security systems" lead by G. Demenko, where the corpus data are required for building a linguistic model for purposes of speech recognition. For that project sensitive texts were anonymized manually: the operator replaced all original names in the document by fictitious ones (Szymański *et al.*, 2009). For the description of machine anonymization (or de-identification) of free-text medical records (in English), see (Neamatullah *et al.*, 2008).

For the needs of corpora-based Machine Translation we need volumes of texts that make manual anonymization too expensive. On the other hand a mechanism for processing NEs seemed necessary to improve the output of the translation algorithm itself. Therefore, we decided to develop a system that would satisfy both needs: Named Entity Recognition for anonymization **and** translation. For the discussion of Machine Translation aspects of our system, see (Graliński *et al.*, 2009).

The paper is organized as follows: In Section 2 we describe the problem of Named Entity Recognition (NER). In Section 3 our grammar for the description of NER rules is provided. We provide some examples of rules compatible with the grammar in Section 4. In Section 5 the rules used in anonymization are discussed. In Section 6 the anonymization tool is described. In Section 7 we evaluate the solution. We end with conclusions and the reference to future work in Section 8.

2 Named Entity Recognition

Named Entity Recognition consists in automatic determination of continuous fragments of texts (called Named Entities) which refer to information units such as persons, geographical locations, names of organizations, dates, percentages, amounts of money, references to documents. A NER system is usually expected to work on a raw text and provide a markup on boundaries and types of included NEs.

Here is an example of such a markup from Mikheev (1999), cited also by Nadeau (2007):

> *On <Date>Jan 13th</Date>, <Person>John Briggs Jr</Person> contacted <Organization>Wonderful Stockbrockers Inc</Organization> in <Location>New York</Location> and instructed them to sell all his shares in <Organization>Acme</Organization>.*

The sixth Message Understanding Conference (MUC-6) (Grishman and Sundheim, 1996) is often considered as a time moment when NER was first recognized as a domain of Natural Language Processing. First attempts in the NER field consisted in creation of handcrafted rules (Rau 1991, Ravin and Wacholder 1996). Then this idea has been driven out by machine learning techniques. Among them are:

- supervised learning — NER process is learned automatically on large text corpora and then supervised by human (Asahara and Matsumoto 2003, Mccallum and Li 2003),

- unsupervised learning — NER process is not supervised; instead existing semantic lexical databases are consulted, such as WordNet (Alfonseca and Manandhar, 2002),
- semi-supervised learning — this "involves a small degree of supervision, such as a set of seeds, for starting the learning process" (Nadeau, 2007).

The survey of NER solutions is well presented by Nadeau and Sekine (2007).

Some research in the field of NER has been done for the Polish language (Piskorski, 2004). The author developed a rule-based formalism for the recognition of NEs from Polish texts and handcrafted a set of NER rules for Polish. However, in our opinion, a complex formalism suggested there makes it difficult for the linguists to create the rules.

The Spejd formalism invented by Przepiórkowski (2008) was intended basically for shallow parsing, Spejd has seemed a promising basis for our work. Our formalism, presented in Section 3, is, in fact, the extension of Spejd. The NER engine has been written from scratch (Spejd engine was not used).

3 NERT grammar

The full formalism for NERT (Named Entity Recognition for Translation) rules is presented in (Graliński *et al.*, 2009). Here we describe a part of it — intended for recognition only:

```
::nert_file::
     (definition)*      # set of definitions
     (rule)+            # non-empty set of rules

::definition::
     Name = pattern     # definition of a pattern

::rule::
     # pattern preceding the match in the same sentence (optional)
     [Before: pattern]
     # left context of the match (optional)
     [Left: pattern]
     # matching text
     Match: pattern
     # right context of the match (optional)
     [Right: pattern]
     # pattern following the match in the same sentence (optional)
     [After: pattern]
     # pattern in the same sentence as the match (optional)
     [Exists: pattern]
     # action invoked if the match is found in the specified context
     Action: action_list

::pattern::
```

```
     # group is a sequence of tokens that meet the same conditions
     group( group)*

::group::
     # a regular expression that may use a NERT definition
     # in brackets
     NertRegExp

::group::
     # set of conditions for the pattern to satisfy
     <condition(,condition)*>
     # number of consecutive tokens that should
     # satisfy the conditions
     ( *|+|{Num(,Num)})*

::condition::
     # orthographical form of the pattern
     orth|
     # canonical form of the pattern being a word or a word phrase
     base
     # matches (or not) a NERT regular expression
     (~|!~)NertRegExp

::condition::
     (
     pos| # part of speech
     case| # case
     num| # number
     gen| # gender
     deg| # degree
     per| # person
     tns| # tense
     sem # semantics
     )
     = Value

::action_list::
     # list of actions which transform source text into target text
     do(, do)*

::do::
     Sem = Value
```

3.1 NERT definitions

NERT definitions aim at simplifying rules by using labels instead of longish expressions, e.g.:

```
# upper Polish letter
UpperPL=[A-ZĄĆĘŁŃÓŚŹŻ]
# lower Polish letter
LowerPL=[a-ząćęłńóśźż]

# Polish word starting with a capital letter
ProperPL={UpperPL}{LowerPL}*

# number token
NUM=[0-9]+
#alternative way of defining a number token
NUM=<orth~[0-9]+>

# regular expression for a zip code
ZIPCODE=[0-9]{2}-[0-9]{3}

#definition of a Polish company suffix
SuffixPL=<orth~(SA|S\.A\.|sa|s\.a\.)>

# Polish month in nominative
MonthPL=(styczeń|luty|marzec|kwiecień|maj|czerwiec|lipiec|
            sierpień|wrzesień|październik|listopad|grudzień)
# Polish month in any case
MonthAnyPL=<base~{MonthPL}>
# Polish month in genitive
MonthGenPL=<base~{MonthPL}, case=gen>

# Polish first name
FirstNamePL=<{ProperPL}, sem=first_name>

PersonPL={FirstNamePL} {ProperPL}
```

3.2 Match part of the rule

Any NERT rule consists of the match part and the action part. The match part consists of the main matching pattern and some optional context patterns:

Before: ***pattern*** Imposes the conditions on the context preceding the match in the same sentence – directly or indirectly.

Left: ***pattern*** Imposes the conditions on the context preceding the match directly, in the same sentence.

Match: ***pattern*** The main matching pattern.

Right: ***pattern*** Imposes the conditions on the context following the match directly, in the same sentence.

After: ***pattern*** Imposes the conditions on the context following the match in the same sentence – directly or indirectly.

Exists: ***pattern*** Imposes the conditions on the context occurring anywhere in the same sentence.

Here are some examples showing how the match part of the rule may be used.

```
# either "Mr." or "Mrs." precedes the text
Left: <orth~(Mr\.|Mrs\.)>

# an inflected form of the words "pan" or "pani" (=resp. "Mr./Mrs.")
# precedes the match
Left: <base~(pan|pani)>

# the text "a.m." follows the match
Right: a\.m\.

# there is a zip code somewhere in the sentence
Exists: {ZipCode}
```

3.3 Action part of the rule

In the rules intended for translation the action part defines the translation for the recognized named entity, see (Graliński *et al.*, 2009). In the rules intended for anonymization, the action part sets the type of the recognized entity.

4 Examples of NERT rules

4.1 Corporation recognition rules

Some named entities denoting corporations may be recognized by their specific endings, such as *S.A.* (*spółka akcyjna* = *joint-stock company*). A simple rule may look like this:

```
Match: {ProperPL} S\.A\.
Action: sem=company
```

A multi-word company name can be recognized as well – `+` should be simply added:

```
Match: <{ProperPL}>+ S\.A\.
Action: sem=company
```

4.2 Temporal expressions

Some examples of temporal expressions, and the NERT rules that process them are presented here:

Sample Polish expression (*English translation*)	NERT rule
1 kwartale 2008r *(1st quarter of 2008)*	`Match: 1 <base~kwartał> [0-9]{4}r` `Action: sem=date`
4 kw 2010 *(4th quarter of 2010)*	`Match: 4 kw [0-9]{4}` `Action: sem=date`
lutego 1986r. *(February 1986)*	`<base~{MonthPL}> [0-9]{4}r\.` `Action: sem=date`
1 czerwca 2007 r. *(June 1, 2007)*	`[0-9]{1,2} <base~{MonthPL}> [0-9]{4} r\.` `Action: sem=date`

4.3 Use of lexicons

In order to distinguish proper names that should be anonymized from common upper-cased words we use lexicons. We have at our disposal a list of Polish first names, a list of most popular surnames as well as a list of cities and locations (20353 entries together with their inflected forms). Exemplary rules that use the lexicon look like this:

```
Left: {FirstNamePL}
Match: {ProperPL}
Action: sem=surname
```

The `sem=first_name` expression in the `FirstNamePL` definition (see Section 3.1) refers to the lexicon information.

5 Recognition Rules

5.1 Creation

The set of rules contains 74 rules recognizing 23 types of entities. It was developed on the basis of 11 documents provided by the local Police Department. Beforehand, the documents were manually anonymized by replacing all sensitive data with fictitious data (rather than tags).

5.2 Usage

The result of applying a rule is twofold: (1) the semantic information is attached to the recognized entity in the context defined by the rule and (2) the other occurrences of the recognized entity are marked.

For instance, the following rule recognizes a company name that is preceded and followed by other company names:

```
Left  : <sem=company> ,
Match : {ProperPL}
```

```
Right : i <sem=company>
Action: sem=company
```

A sample text that present the usage of this rule:

> Pan Jan Kowalski pracował wcześniej w 3 firmach: `@COMPANY@`, Polwax i `@COMPANY@`. W firmie `@COMPANY@` i Polwax pełnił funkcję prezesa.
>
> (Eng. *Mr. Jan Kowalski was previously working in three companies:* `@COMPANY@`*, Polwax and* `@COMPANY@`*. In* `@COMPANY@` *and Polwax he was taking the position of chairman.*)

(The types of anonymized NEs are written using upper-case letters between `@` characters. As it can be seen, two of three company names have already been recognized and marked up.)

In the first sentence the rule will recognize "Polwax" as a company so in both sentences the text "Polwax" will be replaced with `@COMPANY@`. The result of applying the rule will be following:

> Pan Jan Kowalski pracował wcześniej w 3 firmach: `@COMPANY@`, `@COMPANY@` i `@COMPANY@`. W firmie `@COMPANY@` i `@COMPANY@` pełnił funkcję prezesa.

5.3 Types

The rules can be divided into two groups according to the number of times they can be run within one document:

5.3.1 1-run Rules

1-run rules are always run just once for every document because the rule conditions do not refer to semantic information. This group contains 64 rules.

5.3.2 Multiple-run Rules

Multiple-run rules can be run more than once. The rule conditions refer to semantic information which can be changed after applying another rules. When new semantic information is attached the rules are applied again because after the change the rule condition might be met. This group contains 10 rules.

Sample multiple-run rule which recognizes a last name preceded by a recognized last name and the conjunction "i" (Eng. *and*):

```
Left  : <sem=surname> i
Match : {ProperPL}
Action: sem=surname
```

In sample text "Kowalski i Nowak" above rule will recognize "Nowak" as a last name only if "Kowalski" is marked also as a last name. Let us assume that the following text appears in another part of the document: "Pan Jan Kowalski" and that "Jan" is recognized as a first name (from the lexicon) so the fragment can be

marked up as "Pan `@FIRST_NAME@` Kowalski". In that context, following rule can recognize "Kowalski" as a last name:

```
Left  : <base=pan> <sem=first_name>
Match : {ProperPL}
Action: sem=surname
```

From now, the string "Kowalski" is recognized as a last name and all the occurrences of "Kowalski"[2] will be replaced with `@SURNAME@`. In particular, fragment "Kowalski i Nowak" will be marked up as "`@SURNAME@` i Nowak". After applying once again the first rule, the text "Nowak" will be marked as a last name. Also other occurrences of that last name will be marked in the document ("Nowak i Głowacki" to "`@SURNAME@` i Głowacki").

The multiple-run rules are applied unless no new semantic information is attached to the document.

6 Implementation

The anonymizing program is implemented as a set of macros for Microsoft Word. The main macro calls a script written in the Python language, which automatically searches for NEs and replaces them with their types and morphological information. For example, the inflected forms of the name *Jan Kowalski*, i.e.: *Jan Kowalski*, *Jana Kowalskiego*, *Janowi Kowalskiemu*, *Janem Kowalskim*, *Janie Kowalskim*) are replaced by strings, respectively, `@PERSON:MoMP@`, `@PERSON:MoDP,MoBP@`, `@PERSON:MoCP@`, `@PERSON:MoNP@`, `@PERSON:MoLP@`, where:

- `Mo` stands for masculine-personal gender,
- `M`, `D`, `B`, `C`, `N`, `L` stand for declension cases,
- `P` stands for singular number,
- morphological descriptions of syncretic forms are separated by commas.

Below are two authentic fragments of texts: one before anonymization, the other after it (the names are fictitious).

> Nadal nie przyznaję się do popełnienia zarzucanych mi czynów, które zostały mi ogłoszone w dniu 13.12.2008r. Pamiętam co wyjaśniałem i chcę zmienić wyjaśnienia. Ja wyjaśniałem, że nie znam chłopaków z Białorusi, ale w tym miejscu wyjaśnię, że ich znam. Eduarda Kalininczego poznałem około rok półtora wcześniej, znam go z Kijowa i z tego, że jest kierowcą.

> Nadal nie przyznaję się do popełnienia zarzucanych mi czynów, które zostały mi ogłoszone w dniu `@DATE@`. Pamiętam co wyjaśniałem i chcę zmienić wyjaśnienia. Ja wyjaśniałem, że nie znam chłopaków z `@COUNTRY:ŻCP,ŻDP,ŻMsP@`, ale w tym miejscu wyjaśnię, że ich znam. Eduarda Kalininczego poznałem około rok półtora wcześniej, znam go z `@CITY:MnDP@` i z tego, że jest kierowcą.

[2] Including the beginnings of sentences.

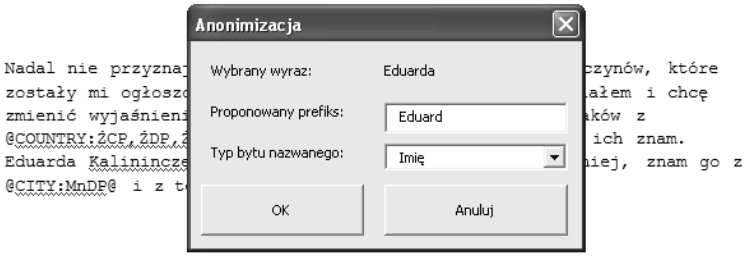

FIGURE 1: Screen shot of the supporting macro.

In case the main macro does not replace all sensitive information, the operator is supported by a set of macros for semi-automatic replacement. In the above fragment the first name *Eduard* (and its inflected forms) are not replaced (this is because the name is not included in the lexicon of Polish first names). To fill up the gap the operator points the cursor upon the word and runs one of the supporting macros (see Figure 1). The operator may choose the type of the entity (*Typ bytu nazwanego*; *Imię* stands for *first name*) as well as the prefix (*Proponowany prefiks*) by deleting last characters from the string. By clicking *OK* the operator demands the program to replace all words starting with the chosen prefix with the selected entity type.

The application is open for use. In case of interest, contact `jassem@amu.edu.pl` or `wawrzyniak@amu.edu.pl`.

7 Evaluation

To evaluate NER systems two measures are referred to most often: recall and precision. Precision is the ratio of the correct guesses to the number of all guesses, recall is the ratio of the correct guesses to the actual number of named entities in the text.

It is not possible for the authors to evaluate the software on original texts, as they contain sensitive information. Therefore we have asked the co-operators from the Police Department, who operate the program, to do the evaluation during anonymization. Specifically, we asked the operators to count the situations when the application does not replace named entities automatically. Having these statistics it has been possible to calculate the recall for each type of NEs.

The precision was not calculated. Firstly, recall is much more important than precision when anonymization is concerned. Secondly, calculating precision would be much more troublesome (as our co-operators from the Police Department would have to confront the anonymized texts with their original versions). Anyway, our further experience with the anonymized texts suggests that lexical units incorrectly recognized as sensitive (and thus anonymized) are rather infrequent (most of them are common nouns written in upper case and misrecognized as identifiers of some sort) and do not seem to be a problem.

The results for 59 Interpol messages (22,506 non-whitespace tokens or 174,289

Table 1: Results for Machine Anonymization of 59 Interpol messages.

NE Type	# All NEs	# Not recognized NEs	Recall (%)
BRAND_NAME	62	0	100.00
CAR	48	0	100.00
CASH_AMOUNT	3	0	100.00
CITY	181	7	96.13
COMPANY	74	4	94.59
COUNTRY	116	0	100.00
DATE	251	0	100.00
FIRST_NAME	340	22	93.53
FRACTION	12	0	100.00
ID	813	0	100.00
ID_CAR	7	0	100.00
ID_PERSONAL	38	0	100.00
LOCATION	37	0	100.00
MONTH	4	0	100.00
NAME	42	0	100.00
NUM	1135	0	100.00
PERSON/SURNAME	344	39	88.66
STREET	47	7	85.11
STREET_FLAT	2	0	100.00
STREET_NUM	19	0	100.00
UPPER_CASE	47	0	100.00
WEIGHT	1	0	100.00
YEAR	17	0	100.00
Total	3640	79	97.83

characters as counted after the anonymization) are presented in Table 1.

8 Conclusions & future work

Named Entity Recognition has been so far applied to various fields of Natural Language Processing, such as: event detection, question answering, semantic information retrieval, text/web mining, machine translation. Here, we present another application for NER: Machine Anonymization. The machine anonymization

program, which has been developed for use in the Polish Platform for Homeland Security project, is open for use.

At the moment the program deals with named entities for Polish texts. The near future work will focus on developing the rules for other languages. The primary goal for the research will be the application in Machine Translation (from foreign languages into Polish). The rules developed there, slightly rebuilt (the action part limited to setting the type of NE), may be used for recognizing NEs in Machine Anonymization of texts written in languages other than Polish.

References

Enrique Alfonseca and Suresh Manandhar (2002), An unsupervised method for general named entity recognition and automated concept discovery, in *In: Proceedings of the 1 st International Conference on General WordNet*.

Masayuki Asahara and Yuji Matsumoto (2003), Japanese Named Entity Extraction with Redundant Morphological Analysis, in *HLT-NAACL*.

Filip Graliński, Krzysztof Jassem, and Michał Marcińczuk (2009), An Environment for Named Entity Recognition and Translation, in *Proceedings of 13th Annual Meeting of the European Association for Machine Translation*, to appear.

Ralph Grishman and Beth Sundheim (1996), Message Understanding Conference-6: a brief history, in *Proceedings of the 16th conference on Computational linguistics*, pp. 466–471, Association for Computational Linguistics, Morristown, NJ, USA, doi: http://dx.doi.org/10.3115/992628.992709.

Andrew Mccallum and Wei Li (2003), Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons, in *Proc. Conference on Computational Natural Language Learning*.

Andrei Mikheev (1999), A knowledge-free method for capitalized word disambiguation, in *In 37th Annual Meeting of the Association for Compuational Linguistics*, pp. 159–166.

David Nadeau (2007), *Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision*, Ph.D. thesis, University of Ottawa, URL `http://cogprints.org/5859/`.

David Nadeau and Satoshi Sekine (2007), A survey of named entity recognition and classification, *Linguisticae Investigationes*, 30(1):3–26, ISSN 0378-4169, URL `http://www.ingentaconnect.com/content/jbp/li/2007/00000030/00000001/art00002`.

Ishna Neamatullah, Margaret M. Douglass, Li wei H Lehman, Andrew Reisner, Mauricio Villarroel, William J. Long, Peter Szolovits, George B. Moody, Roger G. Mark, and Gari D. Clifford (2008), Automated de-identification of free-text medical records, *BMC Medical Informatics and Decision Making*, 8.

Jakub Piskorski (2004), Named-Entity Recognition for Polish with SProUT, in Leonard Bolc, Zbigniew Michalewicz, and Toyoaki Nishida, editors, *IMTCI*, volume 3490 of *Lecture Notes in Computer Science*, pp. 122–133, Springer, ISBN 3-540-29035-4.

Adam Przepiórkowski (2008), *Powierzchniowe przetwarzanie języka polskiego*, Akademicka Oficyna Wydawnicza EXIT, Warszawa.

Lisa F. Rau (1991), Extracting Company Names from Text, in *Proc. Conference on Artificial Intelligence Applications of IEEE*.

Yael Ravin and N. Wacholder (1996), Extracting Names from Natural-Language Text. IBM ResearchReport RC 2033, Technical report, IBM.

Marcin Szymański, Jerzy Ogórkiewicz, Marek Lange, Katarzyna Klessa, Stefan Grocholewski, and Grażyna Demenko (2009), First evaluation of Polish LVCSR acoustic models obtained from the JURISDIC database, *Speech and Language Technology*, 11, to appear.