Anonymization of Court Orders

Claus Povlsen
University of Copenhagen
Centre for Language Technology
DK-2300 Copenhagen
cpovlsen@hum.ku.dk

Dorte H. Hansen
University of Copenhagen
Centre for Language Technology
DK-2300 Copenhagen
dorteh@hum.ku.dk

Bart Jongejan
University of Copenhagen
Centre for Language Technology
DK-2300 Copenhagen
bartj@ hum.ku.dk

Bo Krantz Simonsen
Progresso
http://progresso.dk
bo@progresso.dk

Abstract — **We describe an anonymization tool that was commissioned by and specified together with Schultz, a publishing company specialized in Danish law related publications. Unavailability of training data and the need to guarantee compliance with pre-existing anonymization guidelines forced us to implement a tool using manually crafted rules. We used Bracmat, a programming language that is specialized in transforming tree data structures, to meet the requirement to pass the XML structure of the input document unscathed through the whole workflow. The tool attains a reassuringly good recall, makes almost no chunk errors and reduces the found entity designators to a nearly correct set of entities that the input text refers to, minimizing the time needed for manual check and post-editing.**

Keywords – NER; real life application; high recall rate; consistent assignment.

I. INTRODUCTION

Services that offer users the possibility to make queries in collections of court orders are an important business area in Denmark. Court orders must be anonymized before they can be made publicly available since they contain confidential material.

Originally this anonymization task was conducted manually at the private company Schultz[1] in a costly and tedious process. This process consists of three steps: each segment with confidential content has to be marked with a predefined tag (Named Entity Recognition, NER), while making sure that a given entity (person, company, etc.) is assigned the same tag throughout the document (entity linking, here called Named Entity Normalization, NEN). In the final step the found entity names are replaced by the tags.

Assuming that 100 % recall of confidential named entities could be reached using language technology, the anonymization process would be alleviated enormously by the automated process that Schultz asked us to develop.

In section 2 we describe the NER process that precedes the NEN and de-identification processes that generate chains of anonymized entities. These are discussed in section 3. In section 4 we describe the post-editing tool for the last manual checks. In section 5 we evaluate the anonymization tool and in section 6 we offer some concluding remarks.

II. NAMED ENTITY RECOGNITION (NER)

When the project started, no NER software existed that could find and mark all anonymization candidates in Danish court orders. Neither was it possible to obtain a corpus with which trainable NER-software could be trained. Therefore we had to take refuge to an adaptation of an already developed rule-based NER.

After the Nordic Nomen Nescio Project 2001-2003 [3], one of the partners, Centre for Language Technology, developed the shared algorithms for NER software further in the IDANNA project[2]. We have adapted the IDANNA NER software to the domain of Danish court orders with the set aim to fully comply with Schultz' guidelines for manual anonymization.

The NER module, which is written in Perl, uses manually created grammar rules to identify Named Entities (NEs) in a flat text version of the input document. The rules assign categories as defined by Schultz to each identified NE.

Besides handcrafted grammar rules, the NER module uses gazetteers, lists of stop words and lists of domain specific NEs to attain a high level of confidence. A set of heuristic rules sweeps up some more NEs, but also many false hits. On the basis of these rules, lists and a number of features (e.g., near context, pre- or postfixes and capitalisation) numerical expressions and names are identified and classified. A confidence score from 1 (high confidence) to 3 (low confidence) is assigned to each NE:

```
<anonym cat="pers" certain="1">
Anders Rasmussen</anonym>
```

[1] http://www.schultz.dk/

[2] http://cst.ku.dk/idanna

<anonym cat="case" certain="2">
C-583/97</anonym>
<anonym cat="div" certain="3">
West Side</anonym>

We used 18 NE categories to reflect all domain specific distinctions in the anonymization guidelines.

III. Named Entity normalization (NEN) and De-identification

The main task of the NEN step is to identify and label the referents of the names that have been found by the NER module and to correct as many chunking and classification errors as possible.

In contrast to the NER module, the NEN module must throughout the process operate on the rich version of the text, which is an XML document. Not sharing a measure for text position with the NER module, the NEN module must find each NE anew, using a list of all NEs found by the NER module as guidance. Because the NEN process involves navigating in a tree data structure as well as in character strings, we chose to write the NEN module in Bracmat[3], a programming language specialized in handling complex, heterogeneous data structures by means of pattern matching and expression composition.

First each NE found by the NER module is tokenized. A list of all types occurring in the NEs found by the NER module is then created, each type accompanied by the set of NE candidates that contain that type, see Table 1.

TABLE I. List of types occuring in NEs

Type	*NE candidates*	*Category*	*Score (lower is better)*
Bilka	Bilka	case	1
Bilkas	Bilkas	div	3
Köl	LB Köl	div	2
	LB Köl Bank	finance	1
Lars	Lars Kold	pers	1
Kold	Lars Kold	pers	1

All text locations where a type (see first column) occurs, either exactly or in a variant that is not too dissimilar[4], are annotated with the set of NE candidates (see second column) that belongs to that type. The same text location can be annotated several times, as illustrated in Fig. 1.

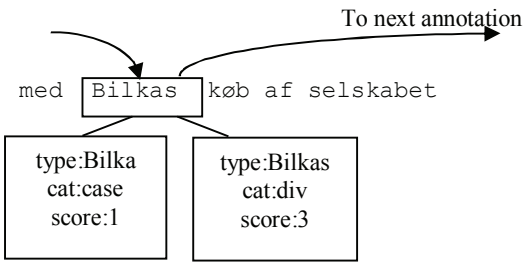

Figure 1. The word 'Bilkas' annotated with two candidate readings. (Translated into English the Danish text fragment reads: "with Bilka's purchase of the company".)

The annotations are all linked together. The NEN module uses the linked list to chunk the NEs, with the possibility to jump over e.g., XML tags, which are not part of the text.

After this initial annotation step, far too many single words are annotated with far too many NE candidates. A sequence of chunking and pruning steps reduces the number of annotations and selects a single NE candidate per annotation.

Many of the NE candidates consist of more than one word (see second column in Table 1), but are nevertheless assigned to single words in the text. The first step is to remove those NE candidates that contain words that do not actually occur in the context of the annotation. To also catch entities that were not found by the NER module, we allow some intervening words between two words in a name to still count as an instance of the name. For example, in *Lars K. Kold*. Both *Lars* and *Kold* have the NE *Lars Kold* as candidate annotation. This candidate survives, because *Lars* and *Kold* are in each other's proximity. Eventually, for this candidate a new chunk is created that comprises all three tokens *Lars*, *K.* and *Kold*.

The next pruning steps are applied until each word with multiple NE candidates is disambiguated. Only keep the NE candidates

- with the most elements,
- with the highest confidence
- with the highest frequency in the text
- that are most qualified by context words like *Inc*. and *Ltd.*
- that occur exactly as in the text.

After pruning, the NEN module attempts to reduce the number of referents by co-reference resolution. For example, the module discovers that Public Service of Mordor and PSMo refer to the same entity when it sees the two names in conjunction, sharing all letters contained in the short name, as in *Public Service of Mordor (PSMo)*.

After co-reference resolution, each surviving name designates a unique referent. Entity normalization proceeds by constructing referent identifiers according to the anonymization guidelines and annotating each occurrence of a name with its corresponding referent identifier. These referent identifiers are used later to de-identify the entities.

[3] https://github.com/BartJongejan/Bracmat
[4] Using the Ratcliff/Obershelp pattern-matching algorithm

IV. POST-EDITING

Once the automatic de-identification has been performed the output is submitted to human post-editing. For this purpose the editor needs a fast and easy-to-use tool that presents key information and provides features for performing common editing operations. The output from the post-editing tool is a fully anonymized XML document.

The main window of the post-editing tool has two panels:

• Panel A contains the document text with highlighted anonymization candidates. This panel provides editing options within the document text and focuses on single occurrences of entities in context, but optionally any changes are ported to all occurrences of the same entity. New entities can be marked-up manually. See Fig. 2.

• Panel B provides an overview that allows the editor to access and edit the same NE simultaneously throughout the document, which often spans many pages.

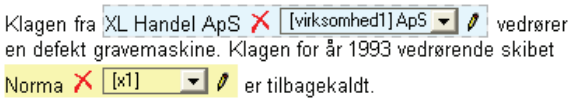

Figure 2. Panel A example (extract). The highlight colour represents a three-level indication of estimated certainty of the automatic de-identification.The colours make it easy to spot where post-editing is most likely needed.

V. EVALUATION

We evaluated the anonymization tool with a document of 16 pages kindly provided by the customer, Schultz, who characterized the document as representative for the texts in the domain for which the tool was developed. Besides precision and recall, we measured chunking quality of the NER module alone and of the workflow consisting of both NER and NEN. We also looked at the effect of disregarding the NEs that are deemed most uncertain by the NER software. See Table 1.

Striving for 100 % recall comes at a price. Not pruning the NER results means that more false positives are found. Our NEN process creates even more false positives, because it searches for all occurrences of found candidate NEs, right and wrong, and may find occurrences that were not found by the NER module, in addition to all occurrences that the NER module did find. However, the 46 false positives when using the combination of NER+NEN were distributed over just 25 false entity designators, the same number as with NER alone. So while the precision nominally decreases relative to the NER process alone, the required post-editing effort to remove the false positives remains the same, requiring 25 actions to remove all false entity designators.

The attribution of categories is improved by the NEN process. While the NER module, lacking a global view of the document, ascribes categories inconsistently and with differing confidence levels, the NEN module, when resolving conflicting categories, keeps the category with the highest confidence level.

TABLE II. TEST RESULTS OF NER AND NER+NEN, WITH AND WITHOUT PRUNING

True number of	*tokens*	347			
	designators	35			
	entities	30			
Hits with low confidence are …		Pruned		not pruned	
Workflow steps		NER	NER+NEN	NER	NER+NEN
True positive name *tokens* found		331	344	347	347
Different *designators* among true positive tokens		34	33	37	36
Different *entities* referred to by true positive tokens according to NEN			29		32
False positive *tokens* found		0	0	34	46
Different *designators* among false positive tokens		0	0	25	25
Different *entities* referred to by false positive tokens according to NEN			0		25
Not identified name *tokens*		16	3	0	0
Chunking errors		157	1	157	1
Recall (%)		95.4	99.1	1.0	1.0
Precision (%)		1.0	1.0	91.1	88.3
F-score (%)		97.6	99.6	95.3	93.8

VI. CONCLUDING REMARKS

We have shown that production quality software for NER and NEN, even in the absence of training data, can be implemented using gazetteers and rule based components. We have also shown how the NEN component can improve the quality of the previous workflow step, NER, by first constructing lists of well-corroborated entities and then revisiting the text to exclusively look for these entities, thereby improving the chunking of already found instances and potentially also finding instances that had not been detected by the NER software.

We think that rule based NER is a good solution for commercial production systems working with highly sensitive data, even if training data would have made it feasible to train a statistical NER system. A rule based system can be made to adhere to business logic that the commercial partner prescribes. It is hard or impossible to prove that a statistically based system can meet this criterion and attain a recall close to 100%. Various investigations imply that even though very high F1-scores can be achieved, it appears difficult for such systems to come near a 100 % recall [2], [5], and [6].

REFERENCES

[1] "Four steps to making data masking a reality". Data Masking Best Practice, White Paper issued by Camouflage Software Inc: https://datamasking.com/wp-content/uploads/2015/05/Data-Masking-Best-Practice.pdf, 2013.

[2] O. Ferrandez, B. R. Scott, S. Shen, and S. M. Meystre. "A Hybrid Stepwise Approach for De-identifying Person Names in Clinical Documents", in Proceedings of the 2012 Workshop on Biomedical Natural Language Processing.

[3] J. B. Johannessen, K. Hagen, Å. Haaland, D. Kokkinakis, P. Meurer, E. Bick, Dorte H. Hansen, A. B. Jónsdottir, and A. Nøklestad. "Named Entity Recognition for the Mainland Scandinavian Languages". Literary and Linguistic Computing, Volume 20, Issue I, 2004, pp. 91-102.

[4] H. H. Olesen. "Rapport om etablering af en offentlig domsdatabase", Domstolsstyrelsen, 2004.

[5] D. C. Svendsen-Tune, "Now you see me, Now you don't – Automatic De-Identification in Court Ruling Documents", unpublished, 2013.

[6] G. Szarvas, R. Farkas, and R. Busa-Fekete.. State-of-the-art Anonymization of Medical Records Using an Iterative Machine Learning Framework. Journal of the American Medical Informatics Association, Volume 14, issue 5, 2007, pp. 574–580.

[7] S. Vinogradov and A. Pastsyak. "Evaluation of Data Anonymization Tools in Proceedings of the Fourth International Conference on Advances in Databases, Knowledge, and Data Applications, 2012 pp. 163-168.