Research Master Thesis

BERTje-based Automatic Anonymisation of Dutch Police Reports

Aju Shrestha

a thesis submitted in partial fulfilment of the requirements for the degree of

MA Linguistics
(Text Mining)

Vrije Universiteit Amsterdam

Computational Lexicology and Terminology Lab
Department of Language and Communication
Faculty of Humanities

Supervised by: Sophie Arnoult
2^{nd} reader: Isa Maks

Submitted: June 24, 2021

Abstract

Anonymisation is an important tool to strengthen data security within the public and private sector and increasingly in demand with legislations as the GDPR in place. The CBS—The Central bureau for statistics—a Dutch independent governmental organ, has initiated an automatic anonymisation project, as a pre-step to automatic crime-categorisation, for a corpus of close to a million Dutch police reports. These documents contain a plethora of sensitive information and are challenging to navigate due to their non-standard language and form. The texts, therefore, require a language and domain specific context-preserving anonymisation tool. This project is a part of the first explorative stage of creating tools for automatic anonymisation at CBS. This is done with the NLP task of NERC, using state-of-the-art transformer architecture. A dataset of 950 police reports are annotated with 16 labels to fine-tune a BERTje-based model. A total of 56 experiments are run, testing with various hyper-parameter settings, datasets and label merges to choose the best context preserving automatic anonymisation model of Dutch police reports.

Declaration of Authorship

I, Aju Shrestha, declare that this thesis, titled *BERTje-based Automatic Anonymisation of Dutch Police Reports* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: 24 June 2021

Signed:

Acknowledgments

TW: may contain dramá — I am assuming very few people will actually read this.

The process of beginning my internship and writing this thesis was spread out through times of crises, both on a global as on a local and personal level. I have had the absolutely unbelievable, but nevertheless very real experience of living through many unfortunate turns of events in the first-time masters program as well my personal life, only for all that to be outdone by a global pandemic. A period of processing a loss, so close and complex, for the first time in my adult life, while working long hours in isolation and finding coping mechanisms in dealing with the minor and major changes and obstacles, as we are hopefully nearing the end of a third wave. This all was happening during the unfolding of major developments in spaces and movements of intersectionality and decoloniality—which I still feel connected to and that I had quite recently withdrawn from to a significant extent, in, retrospectively very necessary, steps that felt like ruptures, to focus on my masters and manifest minor and more sustainable practises to these ends.

Whichever way you look at it, we have been living through historically significant times. Despite the hurdles I faced to do so, working on the internship and thesis has helped me to stay centred and connected in a time of global disconnect and disorder. I feel grateful to have been able to work on a fragment of a solution to a major and multifaceted matter.

I would not have been able to do it without the support, supervision, and encouragement of many people who have showed up throughout this period of my life.

I would like to start by thanking my supervisor Sophie for her wonderful guidance throughout the process: from learning to work with Transformers models to explaining all the machine learning things that helped me in bridging the knowledge gap due to the unfortunate scheduling-conflict in our year. I am happy to have had a supervisor who has been so helpful, patient, consistent, and understanding. I greatly admire that she was able to do this while raising young children and finishing her PhD in a lockdown.

I am grateful to have been able to do my internship at the CBS and experience a healthy (digital) working environment with two incredibly helpful supervisors, Guido van den Heuvel and Bart Bakker, who have provided consistent guidance throughout the process. I am thankful for Bart's supervisory and academic expertise and Guido's persistent technological guidance in working through the very necessary, yet sometime very tiring restrictions of working in a protected digital environment.

Overall this period has helped me to grow more comfortable in opening up about

my personal struggles and asking for help in resolving challenging situations. Having supervisors at both CLTL and CBS who have provided spaces where I feel safe enough to do so has contributed to recovering my faith in academia.

Lastly, I would like thank my friends and family, some of whom I haven't seen in real life for over a year, who maintain their consistency in checking up on each other from great distances. I will mention only a few who have been remarkably present in a these trying times.
I would like to thank my family, whom I get to share this whirlwind of a diasporic existence with. My parents, who never missed an opportunity to ask about my thesis, for helping out how they could, and who I hope to hug as soon a they are fully vaccinated. Amu, for being a grounding presence in my life and her wonderful hospitality in providing her home as an alternative study place when working from home alone became too exhausting. I would also like to thank her son, Haku for being such a cute cat and teaching many hasty humans that chilling is a very valid and honourable way to pass time. Nugah, for being so political and young, and allowing me to taste your dumplings.
Marielle, for our many walks, talks, and being able to share and celebrate our joys and love for this earth. The Queen of YAS or the YAS Queens (Ali, Fiona, Meret): who's (digital) presence and zoom calls have brought life and hope in times of trouble. Louise, for proofreading a larger part of the work. I am grateful to have been a part of a small, yet supportive group of students of HLT and Text-mining. I would also like thank my plants, who taught me about the circle of life and that what you (sufficiently) water, grows. I look forward to a time where I don't have to look at a screen for a while.

List of Figures

Contents

Chapter 1

Introduction

This project aims to automatically anonymise data from *processen-verbaal*—official written records provided by the Dutch police[1], in order to conceal information that potentially enables individuals to be identified. The research was initiated as a pre-step in automatising cybercrime-classification of police documents at CBS (Centraal Bureau voor Statistiek), to prevent the disclosure of sensitive information of individuals and organisations mentioned in these texts.

Furthermore, this is the first step within a larger programme at CBS to develop tools to conceal private information. The intended end result is therefore not a flawless anonymisation tool, but rather the development of a (set of) start-up model(s) part of the first explorative stage.

Motivation

The execution of this project is in alignment with CBS's target to comply with the European Union *General Data Protection Regulation* (GDPR) and the complementary Dutch *Algemene verordening gegevensbescherming*— the AVG-law—that were implemented in May 2016 and apply since May 2018. These regulations function to give individuals more control on how and for what purposes their data is gathered and used by companies and governmental bodies and legally obliges these organisations to justify the storing and usage of this data (van der Sangen, 2018).

Another piece of legislation to which compliance will be optimised is the CBS-law, a Dutch legislation implemented in 2003, granting the institution access to (classified) data for statistical purposes from taxable entities. This is under the condition that necessary technical and organisational measures are taken to protect it from loss, damage and unauthorised examination, alteration and provision as described in section 38 of the CBS-law. The application of automatic anonymisation to the data provided will facilitate in taking these measures.

Additionally, the project contributes to strengthening CBS' core values to safeguard the private information of citizens and organisations to which it has been provided access, in order to maintain its reputation as a trusted governmental institution.

While compliance to the aforementioned laws by CBS has already been satisfied, it will be further refined by limiting the exposure of sensitive data to researchers handling

[1]Throughout this work, the documents may be referred to as *police reports*, *(police) records*, *processenverbaal*, *(official) written records*, *documents* or simply as *the data*.

the documents. The current approach to this matter is by contractually obliging them to keep the sensitive information they have retained strictly confidential. Anonymising the data will hence reduce the risk of disclosure of classified data.

All data accessed for building the anonymisation tool was secured to the best ability and technological capacity of CBS. The documents were strictly made available to the few people who were involved in the project and all persons involved were legally bound to keep the data secure.

A final matter that anonymisation may help resolve is the mis-associative patterns some machine learning models have previously developed in the task of classifying whether a document is cybercrime related. Within an internal CBS research, it was found that certain proper names were erroneously marked as a condition for the cybercrime class in the performance of the algorithm. Replacing such proper names with a generalised class may therefore reduce similar problematic outcomes.

Anonymisation

In the process of anonymising personal information in these texts, Natural Language Processing methods will be applied. Anonymisation is described by Medlock (2006) as follows:

> *Anonymisation is the task of identifying and neutralising sensitive references within a given document or set of documents.*

An inevitable part of the process of anonymisation is the trade-off between concealing sensitive information and preservation of readability and content - described as the *content deterioration dilemma* by Kleinberg et al. (2017). Ideally, in an anonymised output, private data should be replaced in a way that conserves context. Existing works on automatised anonymisation are rarely open-source or context-preserving (Kleinberg et al., 2017). Moreover, these are often shaped towards texts that are of a more formal and regularly structured nature, such as scientific publications (Sweeney; Motwani and Nabar, 2008; Neamatullah et al., 2008; Vico and Calegari, 2015; Kleinberg et al., 2017). To my best knowledge, no open access anonymisation tool for the Dutch language has sofar been made available.

Related works

The language of almost all documents is Dutch; making this project, to our knowledge the first academic work[2] on a context-preserving Dutch anonymisation model, certainly for Dutch police reports. Parallel to this work, Plamondon et al. (2020) are currently developing an anonymisation toolkit with context preserving options for all official EU languages and the work is expected to be completed by December 2021. These models will however be a-tuned for usage in public administration in health and legal domains. Similarly, previous works on automatic anonymisation that are remotely connected to the domain of police records fall within the legal domain, that often contains more formal and regulated language (Kleinberg et al., 2017). The CBS dataset however mainly consists of natural language that is unedited. These texts may not comply to standardised language rules as they are often written under a form of stress. In addition,

[2]Tools for anonymisation may exist within a company or institution for which publications are unavailable.

besides containing non-normative spelling and grammatical structures, the documents occur in various irregular formats.

Method

The task of Named Entity Recognition and Classification (NERC) will be used as a pre-step to detect tokens that require anonymisation. In carrying out of this step, Machine Learning (ML) methods will be used. In order to create models that are tailored to the records, the performance of this task will be preceded by an annotation process. A sample of the data - which consists of nearly a million records - will be annotated in order to create training, validation and test data for the ML models. The desired result of the project is a tool that performs context-preserving anonymisation. In the output, the concealed Named Entity will be replaced by a label which denotes a hypernym it can be categorised under, e.g. *I live in Italy* would be anonymised as: *I live in #LOC* instead.

A Domain and Task Specific NERC-tool

NERC has been carried out for a variety of languages and domains in a collection of studies, resulting in entire subfields and surveys. The task itself will be further elaborated on, in the upcoming chapter. Nevertheless, in preparation for this project, it was determined that creating a domain-specific tool is necessary. The creation of a specific NERC-tool for the data and the subtask is required due to a number of reasons. I will describe these in the following:

Domain-specific labels
One of the main objectives is that the task necessitates selective labelling as the anonymisation of the data precedes the binary categorisation of each document as cybercrime or not cybercrime. This follow-up task requires any information indicative of the document to be classifiable under the cybercrime category to be preserved. To resolve this matter, labels specifically tailored to cyber-related NEs are created[3]. At a later stage one could opt to keep or discard these labels for the anonymisation; when preserving these labels they could function as an indicator of cybercrime while concealing sensitive information.

Selective labelling additionally benefits the execution of the task as domain specific labels can be added. The written records contain domain specific entities that require anonymisation such as licence plate numbering.

Writing styles and ineffectiveness of readily available tools
The data are *processen-verbaal* and consist mainly of unedited natural language text. They may be written under (time) pressure by writers who do not necessarily prescribe to standardised language rules. The data contain jargonic, telegraphic, and other non-normative spelling and grammatical varieties and structures. Moreover, the documents occur in irregular formats; portions of the texts occur in the shape of filled-in forms, hence greatly varying from natural language structure and even containing discontinuous NEs (e.g. street name and house number are separated by other tokens). This

[3]See the Annotation Guidelines in Appendix A for all semantic classes, including these cyber-labels

expectedly results in ineffective application of existing NERC-tools, even when merely implemented to reduce the amount of work for annotators by providing pre-annotated tokens.

While there are various readily available NERC programs, within the protected CBS-environment only the spaCy NERC-tool *EntityRecogniser*[4] was operative for Dutch. In testing the tagger on a sample of the data, it was quickly established to be flawed and inadequate even as a pre-tagger to facilitate the work of annotators. Besides not picking up on most of the NEs present, it additionally mislabelled non-NEs tokens as such, therefore it may have potentially burdened the annotator with more workload rather than relieve them of it.

Limitations

Among the limitations of using NERC for anonymisation are that (1) not all NEs require anonymisation and (2) not all sensitive references are named entities e.g. detailed descriptions of suspects. The latter remain visible within the scope of this project as their presence is very limited within our sample of the data. The potential problem of over-anonymisation that could result from the former limitation, is circumvented by giving non-sensitive NEs distinct labels in the process of NERC. These labels can subsequently be discarded in the anonymisation process, where only specific token sequences are replaced with a context-preserving anonymisation label.

Research questions

The research questions that this project aim to answer are:

1. (How) can (an) NERC-based anonymisation tool(s) be applied effectively to the domain of police reports?

2. Is it possible to create one anonymisation tool for a variety of writing styles and text structures?

Plan Outline

These questions are examined in the following chapters. In the next chapter, the task of NERC is elaborated on further in section 2.1, available models are discussed in section 2.2, and a system is chosen to execute the task in 2.3. In chapter 3, the creation of the dataset is described. In the subsequent and fourth chapter, the chosen model is experimented with and the results of these runs are reported. In the final chapter, conclusions are drawn from the results, they are further discussed, and ultimately recommendations are given to CBS for the continuation of this project.

[4] https://spacy.io/api/entityrecognizer

Chapter 2

Named Entity Recognition and Classification

In this chapter, I discuss the NLP task of Named Entity Recognition and choose a method to apply this task for the anonymisation process. I start by describing the task and its origins in section 2.1; I discuss the methods available in 2.2; and finally, in section 2.3, I choose the method most suited for this project.

2.1 Introduction

The term *Named Entity* (NE) was first coined in the Sixth Message Understanding Conference (MUC-6), where it was defined as *the names of all the people, organisations, and geographic locations* (Grishman and Sundheim, 1996). With the expansion of the field, this definition has since transformed; however the description of the NEs as *proper names*, while not always fitting, has accumulated saliency. These (proper)names are given independently of common characteristics (Benikova et al., 2014), for example: two people may both be called *Audre*, this however does not insinuate any connection or similarity between these persons. In contrast, common nouns such as *doctor* or *swimmer* denote a generic class or group and do not identify specific entities.

The task of Named Entity Recognition and Classification (NERC), in practical terms, is a two-step process consisting of: 1) Named Entity Detection (NED) or Recognition (NER): detection of the tokens that belong to named entities; 2) Named Entity Classification (NEC): assigning these named entities to semantic categories (Benikova et al., 2014). The labels assigned are domain specific and can vary greatly from context to context. The most commonly used are PER (person), LOC (location), ORG (organisation), and MISC (miscellaneous)(Nadeau and Sekine, 2007). They are also referred to as semantic classes or entity types.

Within NLP, NERC is an Information Extraction (IE) task and can be applied for a range of purposes and often functions as a step that is part of a larger or more layered task, such as co-reference resolution or anonymisation.

2.1.1 Background

Historical development

Among the earliest works on the task is a paper by Rau (1991) describing a heuristics and rule-based system to detect and extract company names. She proposes this tool to tackle a major problem in the field of NLP at the time: the presence of unknown words in the form of names. It was not until MUC-6, in 1996, that the phrase *Named Entity* was coined in the context of IE tasks that aim to extract structured information of company activities and defence related activities from unstructured text (Grishman and Sundheim, 1996). Ever-since, the task has been carried out for a variety of languages and domains in a collection of studies resulting in entire subfields and surveys. Among the more prominent and comprehensive surveys on NERC is the one by Nadeau and Sekine (2007), the first one of its kind. In section 2.2, I touch upon approaches to the task that were presented in their overview. Complimentary to this work, I mention methods included in a survey on deep learning NERC models by Yadav and Bethard (2019), as well as models based on transformer architecture (Vaswani et al., 2017).

Language, Domain, Entity types

Models are shaped by the language of the data. While, unsurprisingly, most of the research has been on English data, Dutch has been represented strongly in the research since the CONLL-2002 conference - as it was one of the only two languages in which the datasets were presented, alongside Spanish, resulting in an upsurge in NERC research on the language (Sang, 2002; Nadeau and Sekine, 2007).

Another foundational aspect of a language system is the domain and genre of the data input. Domains are the specific field of the information of the data (e.g. legal, business, biomedical, gardening) and textual genre is the writing style (e.g. scientific, journalistic, informal) (Nadeau and Sekine, 2007).

The latter has shown to greatly affect research results. In a work by Poibeau and Kosseim (2001) several models were tested on two corpora with each a different textual genre, namely: (1) newswire texts, and (2) transcriptions of phone conversations and technical emails. These experiments resulted in differences of precision and recall of up to 40% (Nadeau and Sekine, 2007).

The specific entity types the model learns to classify is dependent on the context and specific goal with which a model is built. The classification label set for a system can range from a small general label set with only persons, locations and organisations classified to domain specific labelling, such as binary protein recognition (Tsuruoka and Tsujii, 2003), to unsupervised "open domain" NERC where ontologies are automatically extended (Alfonseca and Manandhar, 2002).

2.2 Methods

In the following I briefly discuss the available methods for NERC. I first touch upon knowledge and rule-based systems and subsequently discuss semi-supervised and bootstrapping systems. I then, expand on unsupervised models, followed by supervised machine learning systems with features. Hereafter, neural network (NN) based methods are mentioned and finally, transformer architectures are described.

2.2.1 Knowledge based models

Earlier NERC models were knowledge-based. Systems that are rule-based and/or based on lexicons and gazetteers, perform effectively when they are extensive and when their composition matches the domain that models are applied for. They underperform however when the input dataset contains NEs that have not been included in the lexicon. The precision in these systems is often high, while the recall is low when domain and language specific rules are inadequate and lexicons are not exhaustive. Therefore applying these methods result in any NEs not covered by their implementation to be left unlabelled. Furthermore the dictionaries these systems are based on are often labour-intensive. They require maintenance from domain experts, as over time they become outdated when new developments in the field occur (Yadav and Bethard, 2019). However, rule-based systems are preferable when training data are scarce or lacking. Algorithmic methods such as semi-supervised and unsupervised models can be applied as well in such circumstances.

2.2.2 Unsupervised and semi-supervised bootstrapping systems

Semi-supervised and bootstrapped systems rely only on a tiny portion of training data. These models rely for example on a set of seeds, i.e. a small number of sentences, with the NEs labelled, are given as input and used to detect sentences containing these NEs in the dataset and their contextual information is used to retrieve and label more NEs. This process is continuously repeated on the growing set of labelled NEs (Nadeau and Sekine, 2007). Among the methods applied are a system by Collins and Singer (1999) where only the seeds were labelled and that relied on 7 features such as capitalisation.

Unsupervised systems are based on clustering to form groups of potential NEs. This can be done based on the context of NEs, lexical patterns or lexical resources such as WordNet[1]. The methods combine the pattern finding aspect of a rule-based systems and feature engineering with a grouping algorithm (Nadeau and Sekine, 2007; Yadav and Bethard, 2019).

2.2.3 Supervised ML systems I: Feature-based non-NN models

In feature-based supervised models, large datasets with vectorised features of each token (input) and their class (expected output)—whether it is a NE and which NE—are given as training data to a probabilistic ML model. With this input, it learns to distinguish NEs in unseen data. The trained model can be examined through a set of runs on a test dataset. Its results are then compared to human annotated data to evaluate the model. Varieties of feature sets can be used during the training phase to optimise the system.

This approach to the task involves feature engineering where useful features are defined and created. Features such as the part-of-speech(POS) tag of a token, orthographic features (often binary) as whether it contains upper case characters or digits, and the lemma of the token are commonly used in NERC. Beside these, information about the preceding and succeeding tokens or the dependency structure can be encoded. Additionally, a binary feature of whether a token(sequence) is included in gazetteers may be added. Word embeddings can be used as well as a pre-trained feature. These

[1] https://wordnet.princeton.edu/

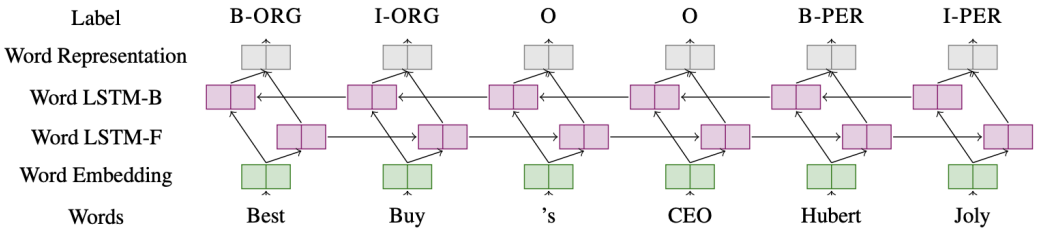

Figure 2.1: Bi-LSTM with word-level architecture from Yadav and Bethard (2019).

are trained with a large unlabelled dataset and their vectors represent the relation between words in the dataset (Collobert et al., 2011; Mikolov et al., 2013; Yadav and Bethard, 2019).

The ML models that have often been used for this task are Hidden Markov Models (HMM) (Bikel et al., 1997), Support Vector Machine (SVM) (Asahara and Matsumoto, 2003) and Conditional Random Field (CRF) (McCallum and Li, 2003).

A high quality large annotated training set, a ML system based approach and the optimal features set to train a model are the basic requirement to construct feature-based supervised classifiers.

Feature engineering has declined in popularity as it is time and labour intensive and these methods were eventually outperformed by models based on a neural network architecture, that do not require feature vectors as an input.

2.2.4 Supervised ML systems II: Deep-learning models

While the first neural network based models for NERC continued to rely on feature vectors(Pascanu et al., 2013), this structure was eventually replaced by architectures where word and/or character inputs are encoded into embeddings. However, successful experiments with embeddings in combination with other features such as POS, case and CRF have been conducted later on, resulting in improved performance (Shao et al., 2016; Yadav and Bethard, 2019).

The types of NNs are among others, convolutional neural networks (CNN), recurrent neural networks (RNN), Long-short term memory (LSTM) and bidirectional long-shortterm memory (Bi-LSTM). RNNs are a structure that takes sequential input vectors that are processed token by token and information about previous tokens is passed on to the next step through a hidden state of the network. A drawback of the structure is the issue of the vanishing gradients, where the gradient shrinks as the NN back-propagates and the smaller weight barely contribute to the learning process. The system is furthermore biased towards the most recent inputs and therefore learn less from vectors-inputs earlier on in a sequence (Lample et al., 2016).

A step to resolve this issue is through the LSTM architecture. This is implemented through the addition of a memory cell that is guarded by gates that regulate the input and what is discarded from the previous cell state (Hochreiter and Schmidhuber, 1997). Bi-LSTM additionally includes a representation of the context of a sequence in the opposite direction and subsequently concatenates the backward and forward pairs per token, resulting in its bi-directionality(Graves and Schmidhuber, 2005; Lample et al., 2016). See Figures 2.1 and 2.2 for the word and character-level Bi-LSTM representations.

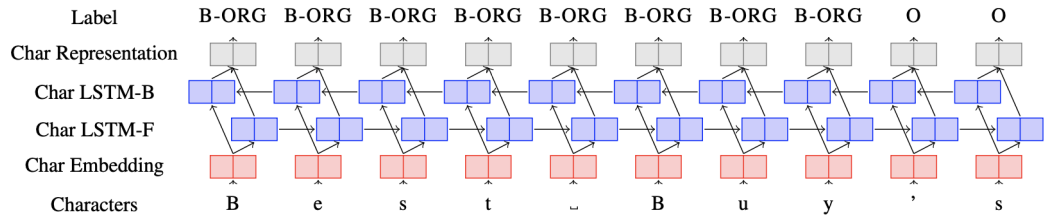

Figure 2.2: Bi-LSTM with character-level architecture from Yadav and Bethard (2019).

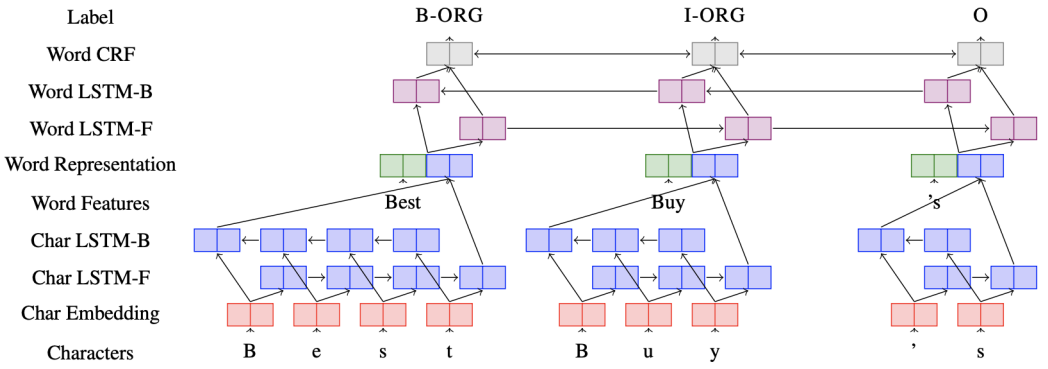

Figure 2.3: Bi-LSTM-CRF with word and character-level architecture from Yadav and Bethard (2019).

The bidirectional-long-short term memory and conditional random field (Bi-LSTM-CRF) with both character and word-level architecture were the former state-of-the-art models. A CRF is used for the tagging process as it is sensitive to patterns of sequence structure in labels and therefore suitable for tasks as sequence-tagged NERC (Yadav and Bethard, 2019; Lample et al., 2016). See Figure 2.3 for a Bi-LSTM-CRF with both character and word-level architecture.

2.2.5 Supervised ML systems III: Transformer based models

The sequential aspect of the former state-of-the-art architecture, Bi-LSTM-CRF, was outperformed by the attention-mechanism of the Transformer model (Vaswani et al., 2017). The attention architecture allows for the data to be processed by the NN simultaneously, rather than in a step-by-step manner. The transformer based pre-trained language model BERT: Bidirectional Encoder Representations from Transformers [2] (Devlin et al., 2019) has resulted in state-of-the-art performances for many NLP tasks (Delobelle et al., 2020). Its realisation is preceded by earlier pre-trained architectures such as ELMo (Peters et al., 2018) and GPT (Radford et al., 2018, 2019). The major restriction of previous pre-trained architectures is their uni-directional quality, i.e. token-level tasks are carried out either by only attending to the previous *or* the succeeding tokens. These structures are particularly unfavourable for tasks that require bi-directional information processing such as question answering and sentence-level tasks (Devlin et al., 2019).

[2] https://github.com/ google-research/bert

BERT employs two steps in its pre-training fase—with the first as the key to realising its bidirectionality:

1. Masked Language Modelling (MLM): during which the model is trained by presenting a sentence with random tokens masked as input for it to output the masked words, giving the model a bi-directional understanding of the language and context within sentence structures;

2. Next Sentence Prediction (NSP): the model is given a set of two sentences to predict whether or not the given sentences occur sequentially, in order to grasp context on a supra-sentential level.

Through these tasks the BERT model is pre-trained and learns the sentence and token (sequence) structures. The tasks are performed simultaneously.

In addition, BERT-models with an optimised pre-training phase have been created such as ALBERT (Lan et al., 2020) and RoBERTa (Liu et al., 2019). For instance, the main novelty of the RoBERTa models is the dropping of the NSP step all together and giving multiple sentences, instead of a single sentence, as input for the MLM procedure during pre-training.

After pre-training, the BERT model can be fine-tuned for specific NLP tasks. This is done with labelled datasets and the adjustment of hyper-parameters. The fine-tuning has achieved impressive results across various tasks, e.g. Question Answering (Devlin et al., 2019) and Text Classification (Sun et al., 2020), even with relatively little data. The strength of the model lies precisely in its ability to excel with small datasets, making it ideal for this project.

BERT Models for Dutch NER

The current state-of-the-art models for many NLP tasks work with the pre-trained language model BERT (Devlin et al., 2019). Since the creation of the model that is pre-trained on English texts, the BooksCorpus and English Wikipedia (cumulatively 3300 million words), variations to the anglophone structure have been created by training it on corpora of one or several other languages. Of these, the models that are most suitable for the task of creating a Dutch NERC system are: (1) the multi-lingual BERT[3] (mBERT), which has been pre-trained on all Wikipedia pages of 104 different languages; (2) BERT-NL[4], the first of three monolingual Dutch BERT models, trained with the Dutch SoNaR-500 corpus (Oostdijk et al., 2013) of 2.2GB as described in Delobelle et al. (2020); (3) monolingual BERTje (de Vries et al., 2019) which is trained on a 5 large Dutch corpora—amounting to a total of 12GB, including one with all the Dutch Wikipedia pages extracted in October 2019; (4) RobBERT(Delobelle et al., 2020), the Dutch RoBERTa (Liu et al., 2019), which has been pre-trained on the Dutch section of the OSCAR Corpus—consisting of over 39GB of texts crawled from the internet (Suárez et al., 2019).

The choice for a suitable model for this project is made in the following selection. RobBERT had unfortunately not yet been released when this project started and could not be considered in the process of choosing the most suitable model. Delobelle et al. (2020) reached near a state-of-the-art performance for NERC with RobBERT v2 (F1:

[3]https://github.com/google-research/bert/blob/master/multilingual.md
[4]textdata.nl

89.01)—their second model, using a Dutch tokeniser for their corpus. These results and features, as well as the impressive corpus it is trained with, are noteworthy and should definitely be considered in future works.

2.3 Selecting a suitable model for Dutch Police Reports

We opted for only one of the models due to limitations in time and chose the monolingual BERTje on ground of: (a) its diversity in the corpora it is pre-trained on, in terms of variety in Dutch texts; and (b) the magnitude of these corpora, which includes the Dutch segment of multilingual BERT and is approximately 8 times the size.

Had time limitations not applied, trying out several systems could be valuable as state-of-the-art results (F1: 90.9, vs 88.3 for BERTje and 89.7 for BERT-NL) for Dutch NERC have been achieved by Wu and Dredze (2019) with mBERT. The previously mentioned results of all the Dutch BERT-models are based on fine-tuning with the Conll-2002 Dutch dataset–consisting of four editions of the Belgian newspaper "De Morgen" from 2000; the newspaper is considered quality press. Considering that mBERT has only been pre-trained on Wikipedia texts, while BERTje has been pre-trained on a more diverse range of texts, the latter may be more suitable for a dataset of filed allegations with a diversity in structure and language usage.

The five large corpora BERTje has been pre-trained on include a large corpus of fiction novels and the SoNaR-500 corpus, which includes informal texts such as chats, blogs and SMS (de Vries et al., 2019; Oostdijk et al., 2013). As the fine-tuning of the model will be carried out with texts of various forms, including informal structure—rather than the moderately formal Belgian Dutch of the Conll data—using BERTje may possibly result in a better performance.

Pre-training specificities of BERTje

There are some differences in the pre-training procedures of BERTje in comparison to BERT: (1) BERTje is pre-trained with the sentence order prediction (SOP) objective—which was first employed in ALBERT (Lan et al., 2020), instead of the NSP task. In SOP, the sentences are trained with either the next or the previous sentence, rather than exclusively the next one as in NSP; (2) for 15% of all the tokens the MLM objective is replaced with a strategy of masking WordPiece-sequences that are a part of the same word, rather a single word piece per time, as this method has been found to be an overly simple prediction task (Lan et al., 2020; de Vries et al., 2019).

Chapter 3

A Dutch Police Reports Dataset for NERC

In this chapter, I will elaborate on the process of developing an annotated corpus from the raw dataset of police records. In section 3.1, I start by expanding on the sample design and further selection of the dataset for the annotation process. This will be followed by a dive into the process of annotation in section 3.2 and finally, the process of creating the datasets to build the models is described in 3.3.

3.1 Sample Design

In this section, I begin with a short description of the raw data, the size of the samples I draw from it, and the strata in 3.1.1. This is followed by section 3.1.2, on two of the three bases for the strata: the sections the texts are assigned to in the raw data, and the source of reports. In 3.1.3, the varieties in text format per source and section are given and discussed. An overview of the number of texts per section and source are shown in 3.1.4. A description on the third basis for the strata, a crime labeling system at CBS, is given in 3.1.5. Following a recount as consequence of adding the crime labels to the strata, a new number of total cases is presented in 3.1.6. The process of natural language extraction in order to create a more diverse dataset is described in 3.1.7. Finally the strata are defined and the samples are drawn for each stratum and presented in 3.1.8.

3.1.1 General information on the population

The content of raw data is official written records (*procès-verbaux*) consisting of a total of 992385 cases before preprocessing. Of these, 900 cases were annotated, as this was what was possible within our capacity. An additional 50 cases were utilised to measure Inter-Annotator Agreement (IAA).
Two samples were drawn, each of 500 texts. In the first sample, the portion of each stratum was made to be of equal size where possible, so it is adequately represented.

The data is made available to CBS by the police in CSV format and is ordered into 992385 rows and 24 columns including the index column. The initial interpretation of the data is to read each row as one case as this holds for the majority of the cases. Exceptions are discussed in subsection 3.1.6, where the recounting of the police stratum is elaborated on.

Throughout this section, I define the strata—categories of text in the data that are based on various qualities and are distinguished in order to ensure there is sufficient diversity of texts in the population sample of each dataset. In the process of choosing the strata, labels are created on the basis of certain conditions within the data. This is done on the basis of (1) the source: reports are obtained from one of three sources (2) the section: the column heading it is documented under in the raw data and (3) the SCM-labelling: a crime-classification system used by the CBS. The first two conditions are described in 3.1.2 and the third is described in subsection 3.1.5.

3.1.2 Sections and source conditions

Each case has been obtained from one of the various sources and have collectively been made accessible to CBS for the purpose of cybercrime (sub)categorisation. The cases have been chosen to be labelled into three different generalised source categories of the police reports:

1. LMIO: criminal complaint by victims from Landelijk meldpunt voor Internetoplichting (LMIO) - the national contact point for Internet fraud;
2. non-LMIO complaints: non-LMIO reports that are filled by citizens;
3. police: written cases by police.

The cases were labelled with source categories based on the following conditions:

- LMIO: whether the text contained the word *LMIO*;
- police: whether the "Verklaring" section contained no text, as this section exclusively contains text by citizens in reports from other sources;
- non-LMIO: the remainder of the cases.

In the dataset, the text of each case is additionally sub-divided under three sections:

1. Toelichting: clarification;
2. Bevinding: finding;
3. Verklaring: explanation.

These were considered in the of process of defining the strata and a narrower selection is made in 3.1.4.

3.1.3 Text format and structure

Each section and source category contains texts with differences in format—i.e. writing style as well as structure—that require to be noted with regard to the NERC and anonymisation task. The writing styles can be subdivided into two generalised categories: citizen writing and police writing. These writing styles may both deviate from standardised language usage. The form of citizen writing varies per report and can be informal and ungrammatical. Police writing contains jargonic language in both vocabulary and on the sentence level. The styles are important to distinguish for a NERC system. Abbreviations in police jargons are often capitalised can be confused with NEs.

Additionally, how non-standard grammatical structures are processed by a system that has been pre-trained on texts with standardised language may vary.

The structure of the text can be subdivided into: natural language (NL) and non-natural language (NNL). The NL text concerns written reports and personal accounts of an incident and the NNL text is part of a filled-in form that consists of word (phrases), always less than a sentence. The forms generally have consistent, predicable patterns. The algorithm used for anonymisation has been pre-trained and fine-tuned to recognise NL patterns and contexts. The NNL text therefore needs to be separated from the NL text. NEs that require anonymisation under this subcategory can be disregarded as the filled-in sections almost solely consist of personal information that requires concealment.

The subdivision text formats can hence be outlined as follows:

1. writing styles:
    1. citizen writing
    2. police writing
2. structure:
    1. NL - Natural Language
    2. NNL - Non Natural Language

Tabel 3.1, presents an overview of the differences in text format by source and section. NL and NNL are easily distinguishable and are additionally separated by a "+" in the table. The writing styles are only stated when regarded as clearly distinguishable after examination. To give a more concrete visualisation of these police reports, artificial police reports of each of the strata are provided in Appendix A.

	Toelichting	Bevinding	Verklaring
LMIO	NL: police writing *or* Similar recurring text with name of police employee leading case and unique reference number + Standard text on LMIO	NL	citizen writing *or* NNL: form + NL: Beschrijving part of form: citizen writing
non-LMIO	NNL: form-like mostly caps + NL: police writing	NL: police writing	NL: citizen writing
police	NNL: Mostly caps + NL: police writing	NL: citizen writing / police writing	-

Table 3.1: Text format per source label and section

3.1.4 Population overview I : raw data

The tables below show the number of cases per source label (Table 3.2) and texts by section (Table 3.3). As shown in both tables, the total number of cases are 992385. Per case, there are 900664 (90.76%) text the sections "Toelichting" and "Verklaring",

and 820141 (82.64%), respectively, contain text[1]. As the "Bevinding" section merely has 63414 (6.39%) rows containing text, with irregular content, it will be discarded in the selection process of the dataset.

	number of cases (before recount)	% of total cases
LMIO	40258	4
non-LMIO	779883	79
police	172244	17
total	992385	100

Table 3.2: Number of cases per source label

	total of fields containing text	% cells containing text per section
toelichting	900664	91
bevinding	63414	6
verklaring	820141	83
total	992385	-

Table 3.3: Number of text containing fields per section

3.1.5 Labelling SCM

In addition to the source and case sections, cases can be labelled into crime categorisations: SCM - *Standaardclassificatie Misdrijven* (standard crime classification). This CBS code system[2] has a total of eight main classes for crimes and various sub-classes. The entire dataset has a total of 94 SCM-codes. By far the largest main crime category is *Vermogensmisdrijf* (property related crime). To roughly encapsulate the differences in the records on the condition of crime type, they will be labelled in to two categories: 'Vermogen' and 'Overig' (rest). While this is a simplified division, alternatives may result in needles and groundless over-complication. See Table 3.5 for the total number of cases per SCM-label after the recount—which is elaborated on the following section.

3.1.6 Population overview II: recount politie stratum and total number of cases

During preprocessing all the documents were concatenated into one table based on the SCMcode. As a consequence the number of cases with police source emerged as 159607, lower than earlier as given in Table 3.2. This indicates that a number of cases had been discarded as a result of this concatenation as the SCMcode field was empty.

[1] counted by selecting fields with more than 1 character; some of these field could still contain text that is unintelligible or uncommunicative and may consists of NNL symbols

[2] See the CBS website for a full overview of all the SCM-categories: https://www.cbs.nl/nl-nl/onze-diensten/methoden/classificaties/misdrijven/standaardclassificatie-misdrijven-2010

Upon further inspections, the large majority, 12555 of the 12637, rows were completely empty and had police as source. As the *police* category was labelled as such based on the condition that the *Verklaring* section was empty as stated earlier in 3.1.2, this is unsurprising. For the other 'missing' cases, the three text sections were empty and contained some disordered, arbitrary text in other sections. Given that this is less than 0.01% of the data, these were deselected, bring the recounted total cases to 979748 (see Table 3.4).

	number of cases after recount	% of total cases
LMIO	40258	4
non-LMIO	779883	80
politie	159607	16
TOTAL	979748	100

Table 3.4: Recounted and final number of cases

	number of cases after recount	% of total cases
Vermogen	591977	60
Overig	387771	40
TOTAL	979748	100

Table 3.5: Number of cases per SCM-label

3.1.7 Natural Language extraction

The natural language extraction of the dataset was applied by deselecting NNL and grossly repetitive strings from the data, to further improve the quality and diversity of the samples. This was implemented through two steps: (1) filtering out clearly and consistently occurring NNL texts; (2) examining the most frequently occurring texts. For the second step, the data considered for deselection mainly consisted of standard reoccurring texts, standard wordings of frequently filed cases, NNL symbols and empty lines that do not require anonymisation. With some text strings occurring over a 10000 times, the ultimate decision was made to remove all strings of texts occurring more frequently than 500 times. Employing a clear limit in frequency made the filtering process feasible and nevertheless resulted in a sufficiently diverse sample selection.

3.1.8 Selection of strata and samples

The strata are divided into the 10 categories as presented in (the first column of) Table 3.5. All names of the categories are three-part abbreviations of: (1) the source categories (LMIO, non-LMIO and police) represented in the first position(s); (2) the sections *Verklaring* (as "v") or *Toelichting* (as "t") represented in middle position in lowercase; (3) the SCM labels *Vermogen* (as "V") and *Overig* (as "O") in the final position in uppercase. To give an example, a text from a non-LMIO source, found in the "Toelichting" section and labelled as a "Vermogen" crime type, would be represented

as "LtV". The second column of the table shows the total number of texts per stratum. The following columns illustrate which contain text and which were deselected and hence labelled to contain "no text".

There were two sampling rounds. The first sample contained 500 texts. 50 texts were drawn for each stratum, with the exception of the LtO and LvO strata of which only 12 occur within the entire dataset. These were supplemented with the strata that are relatively most similar: LtV and LvV. The strata are given equal size (when possible) to ensure each stratum is sufficiently represented in the first sample. This first sample was evaluated by manual inspection to assess its diversity. The second sample was drawn based on the insights gained from these processes.

The lessons gained from the process of annotation pointed to the fact that the majority of the texts in LtV category were of similar structures. In the second sample, even after drawing a relatively larger sample this category was equally monotonous and was eventually disregarded for the annotation process as it would have had a negative contribution to diversity of the sample. The lack of LvO category was compensated by drawing a larger LvV sample to balance out with a relatively similar category. The other samples were kept the same size of 50, resulting in a slightly smaller sample of 400 texts for the second round of annotation.

Strata	texts	text	no text	sample 1	sample IAA	sample 2
LtV	40246	6524	33722	89	9	0
LtO	12	12	0	11	1	0
LvV	40246	40244	2	89	9	100
LvO	12	12	0	11	1	0
nLtV	509184	366258	142926	50	5	50
nLtO	270699	253245	17454	50	5	50
nLvV	509184	503176	6008	50	5	50
nLvO	270699	263841	6858	50	5	50
ptV	42547	29987	12560	50	5	50
ptO	117060	87105	29955	50	5	50
total	979748*	743131*	236617*	500	50	400

Table 3.6: Strata and sample selections for first sample and IAA

3.2 Annotation Process

In this section I will describe the annotation process consisting of a description of Inception, the tool used for annotation in 3.2.1, the annotation guidelines in 3.2.2, the annotation project in 3.2.3, and the IAA in 3.2.4. The following section discusses the resulting datasets.

*this represent the total number of cases, not the sum of all the *Verklaring ánd* Toelichting sections displayed above in the table

3.2.1 Inception

In preparation of the annotation process, the raw data in CSV format were viewed, transformed and deselected through python (packages) in Jupyter Notebook[3]. The selected samples were transformed into conll 2003 input using TextToConll[4] and to be given as input to the annotation software, Inception (Klie et al., 2018). The software tool was used to execute the annotation process. It was applied for both the moderation through assigning datasets and projects to each annotator and the general regulation of the process, and naturally in the carrying out of the annotations. It additionally provided the possibility to calculate IAA. The annotated output data was in conll 2002 format with sequence-tagging.

3.2.2 Annotation Guidelines

Annotation guidelines[5] for the project are modelled after the *Annotation Guidelines for Named Entity Annotation* by Benikova et al. (2014) and tailored to the dataset and the specific labels required. It was made available to the annotators in both English and Dutch. Tables with examples and additional documentation were made available to further assist and prepare the annotators. A total of 16 NE labels were created with 14 labels for NE types that require anonymisation and 2 labels to facilitate the process for the machine. The latter 2 types can be disregarded for the final result. The labels and descriptions of their categories are:

1. #PER: proper names denoting persons
2. #USER: virtual platform usernames
3. #ORG: proper names of organisations, companies or institutions
4. #EDU: educational institution
5. #WEBAPP: virtual platform
6. #LOC: places described with a proper name (addresses included)
7. #LOCderiv: derivation of places described with a proper names
8. #DATE: phrases indicating dates (of birth), time units exceeding hours
9. #MAIL: email address
10. #PHONE: phone number
11. #BANKNR: mainly numerical bank account details
12. #KENT: license plate number
13. #PJ: police jargon
14. #MISC: words have the form of NEs, but do not require anonymisation

[3]Jupyter notebooks and scripts can be found on `https://github.com/yellowonder/anonymisation.git`

[4]`https://github.com/cltl/TextToCoNLL`

[5]The full Annotation Guidelines and its supplements can be found in Appendices B, C, and D.

15. #CODE: numeric codes that are not contained in the classes above

16. #OTH: NEs that require anonymisation and are not contained in the labels above

Further explanation and examples can be found in Appendices B, C, and D.

3.2.3 Annotation Project

The annotation team consisted of 4 annotators from CBS. They worked on the task part-time and labelled approximately 25-40 documents per week. This was both necessary and favourable as (1) they could only commit for a specific number of hours to the annotation project and (2) a portion of the police reports described incidents with heavier topics, therefore spreading the workload facilitated the process of creating healthy working conditions.

The annotation process was divided into 2 rounds. The first round was conducted by 3 annotators (AN1, AN2, AN3) and in the second round a fourth annotator (AN4) was added to increase the speed of the process. The annotators completed the annotation of 500 documents in the first round. The documents were distributed among the three annotators with two persons labelling 170 documents each and one person annotating 160 documents. The second round of consisted of a total of 400 documents with two annotators labelling 107 documents each, one annotating 47 documents and the additional fourth annotator labelling 139 documents in total. The distribution of the strata among the annotators in round 1 is displayed in Table 3.7.

For round two the initial distribution of the strata among the first three annotators

Round 1					
		AN1	AN2	AN3	total
1	LtO	4	4	3	11
2	LtV	30	30	29	89
3	LvO	4	4	3	11
4	LvV	30	30	29	89
5	nLtO	17	17	16	50
6	nLtV	17	17	16	50
7	nLvO	17	17	16	50
8	nLvV	17	17	16	50
9	ptO	17	17	16	50
10	ptV	17	17	16	50
	total	170	170	160	500

Table 3.7: Distribution of strata in annotation round 1

was the same, given in Table 3.8. Due to time limitations for one of the annotators (AN3) the distribution was slightly moderated to the format as shown in Table 3.9. The datasets resulting from the annotation rounds are named with the annotator number and round and maintain the initial distribution of Table 3.8 (see section 3.3).

Round 2.1						
		AN1	AN2	AN3	AN4	total
01	LvV	25	25	25	25	100
2	nLtO	12	12	12	14	50
3	nLtV	12	12	12	14	50
4	nLvO	12	12	12	14	50
5	nLvV	12	12	12	14	50
6	ptO	12	12	12	14	50
7	ptV	12	12	12	14	50
	total	97	97	97	109	400

Table 3.8: Initial plan of distribution of strata in annotation round 2

Round 2.2						
		AN1	AN2	AN3	AN4	total
1	LvV	25	25	25	25	100
2	nLtO	12	12	12	14	50
3	nLtV	12	12	10	16	50
4	nLvO	12	12	0	26	50
5	nLvV	12	12	0	26	50
6	ptO	12	20	0	18	50
7	ptV	22	14	0	14	50
	total	107	107	47	139	400

Table 3.9: Distribution of strata in annotation round 2

3.2.4 Inter Annotator Agreement

Inter Annotator Agreement (IAA) measures the consistency in annotations between the annotators. It is measured by having all annotators label the same set of texts and comparing the agreement in the annotations. It was measured after the annotation of 125 documents in the first round for the first three annotators. The fourth annotator labelled the IAA after the completion of 139 of documents. It was measured using 50 documents with a strata distribution as depicted in Table 3.6. The IAA was calculated using Cohen's Kappa coefficient(Cohen, 1960). When κ is larger than 0,80, the agreement can be interpreted as almost perfect. The annotations of each annotator were compared to that of another one. The results of the IAA are displayed in Table 3.10, showing a high degree of agreement between all annotators.

Inter Annotator Agreement				
	AN1	AN2	AN3	AN4
AN1	-	0.92	0.91	0.92
AN2	614/797	-	0.89	0.89
AN3	675/759	598/787	-	0.87
AN4	685/783	606/813	657/785	-

Table 3.10: IAA results using Cohen's Kappa coefficient

3.3 Datasets

The full dataset is divided into three sections for the experiments: the train, test, and development data. The number of annotated NE's of the outputs per round, per annotator as well as the variety of (the total 16) labels are represented in Table 3.11. As the size of the dataset is relatively small, the IAA output has been added as well to increase its volume. The IAA annotations of AN1 are used as these have the highest agreement scores in comparison to the IAA annotations of every other annotator (see first row, Table 3.10). Additionally, the composition in terms of variety in strata of dataset AN3r2 are as given in Table 3.8. The original distribution was maintained, while annotated by all annotators (see Table 3.9). The name AN3r2 for this dataset, is preserved for referential convenience.

The outputs are distributed among the various, train, test and development sets.

	number of NEs	Variety of labels
AN1r1	2995	16
AN2r1	2494	15
AN3r1	2266	16
IAA	730	16
AN1r2	1669	15
AN2r2	2029	16
AN3r2	1944	15
AN4r2	2256	15
total	14439	

Table 3.11: Number of annotated NE's per annotator per round and their label variety

The first two datasets, with the second varying from the first simply in size of training data, are displayed in Table 3.12 and Table 3.13.

Dataset 1				
	training data	test data	dev data	total
number of NEs	10154	2256	2029	14439
composed of	all r1 + IAA + AN1r2	AN4r2	AN2r2	
% of the total	70	16	14	

Table 3.12: Distribution of annotations of rounds and annotators for Dataset 1

Dataset 2				
	training data	test data	dev data	total
number of NEs	12098	2029	2256	16383
composed of	all r1 + IAA + AN1r2 + AN3r2	AN4r2	AN2r2	
% of the total	74	12	14	

Table 3.13: Distribution of annotations of rounds and annotators for Dataset 2

Redistribution strata and rounds

After noting that the LtV stratum was missing from both the test and development datasets in the Datasets 1 and 2, a redistribution of the rounds was applied in order to create an evenly diverse distribution among datasets, in term of strata as well as rounds. The quality of the annotations generally increases with experience, therefore a balance in the rounds per dataset is desired. This resulted in a new distribution of all three datasets in Dataset 3, see Table 3.14, where each round and strata is represented in the training, test and dev data. Finally, Dataset 4 with a larger training set was created, as given in Table 3.15.

Dataset 3				
	training data	test data	dev data	total
number of NEs	7290	4438	2986	14714
composed of	AN1r1 + AN3r1 + AN2r2	AN2r1 + AN3r2	AN4r2 + IAA	
% of the total	50	30	20	

Table 3.14: Distribution of annotations of rounds and annotators for Dataset 3

Dataset 4				
	training data	test data	dev data	total
number of NEs	8959	4438	2986	16383
composed of	AN1r1 + AN3r1 + AN1r2 + AN2r2	AN2r1 + AN3r2	AN4r2 + IAA	
% of the total	55	27	18	

Table 3.15: Distribution of annotations of rounds and annotators for Dataset 4

3.4 Reflection

The annotation process consisted of three steps: (1) the sample design, (2) annotation and (3) dataset composition.

In the sample design natural language texts was filtered out and excessively frequent occurring texts that do not require anonymisation were deselected. This greatly increased the diversity in texts for the drawing of the samples. Another important part of the sample design consisted of choosing the conditions for the strata to be carefully selected, through looking at writing styles and based on the source of a text, the section it was assigned to in the raw data, and a binary and necessary simplification of the SCM code. This resulted in a total of 10 strata with which two samples were drawn: one for the IAA and the first round of annotation, and one for the second round.

Succeeding to the drawing of these samples, the data was annotated by four annotators, all with near perfect IAA scores. Adding a fourth annotator later on in process increased the efficiency time-wise and based on the IAA scores had no visible effect on the quality of the annotations.

Lastly, four datasets were created, with the latter two datasets ensured of an even distributions of strata and annotation rounds as this benefits the experimental process.

Additionally, the second and the fourth datasets were given larger training sets, so the effects of this variation can be tested in the upcoming chapter, where models will be tested with a diverse range of inputs and set-ups.

Chapter 4

A BERTje-based System for NERC in Dutch Police Reports

In this chapter, I discuss the methods and results used to create a BERTje-based NERC model for police reports. The methods and variation are discussed in 4.1. Section 4.1.1 describes the experimental setup through the pre-processing and fine-tuning with hyper-parameters. In section 4.1.2 the experiments with dataset sizes are discussed and in section 4.1.3 label merges are examined and described. Finally, the most suitable model is chosen.

4.1 Methods & Results

The experiments were carried out with a BERTje[1] model downloaded in November 2020. With this model, a total of 56 experiments were run to reach the optimal system with the NERC police reports corpus described in chapter 3. For the experiments, varieties of datasets were created to input and selected hyper-parameters were adjusted. The datasets were first preprocessed, shuffled and converted to the appropriate format. The best model is selected based on the validation data and the results are reported for the test set.

A reduction of false negatives, achieved when recall is high, is desired, as the aim is to achieve a classification model that leaves out as few instances of NEs as possible in labelling—reducing the amount of sensitive data that remains visible. In working towards the suitable model for the task, results scores with a higher recall are therefore preferred over high precision scores. A situation of over-anonymisation is hence favoured over an outcome of under-anonymisation.

The following section, 4.1.1, describes hyperparameter tuning experiments carried out with Dataset 1 as given in Table 3.12 on page 22. In the succeeding section, data-related experiments were run with Datasets 2, 3, and 4 that are displayed in Table 3.13, Table 3.14, and 3.15, respectively. Experiments described in section 4.1.3 on label merges, are all conducted with Dataset 4 (Table 3.15) with labels adjusted to the merge.

4.1.1 Hyperparameter settings

In this section, I describe experiments with the following parameters: whether the input data should be the shuffle, maximum token length per sentence, random seed, batch

[1] `https://huggingface.co/GroNLP/bert-base-dutch-cased`

size, and epoch.

To shuffle or not to shuffle

The shuffling of the data was done with a script[2] that randomly shuffles the sentences with the aid of the python package random and a given random seed. Shuffling the sentences generally makes a system more robust for various contexts, however contextual information of the structures of texts is lost. In the context of our dataset, the structures of the texts are varied, therefore contextual information on sentences sequences may not always be as informative to the system. To examine this matter, an experiment was conducted to check whether shuffling improved the results by running two model with the extract same parameters. The first model was given shuffled input data and for the second run, it was left un-shuffled.

max len	shuffled	seed	epochs	batch size	eval steps	loss	P	R	F1
128	Yes	1	3	16	250	**0.180**	0.645	**0.658**	**0.652**
128	No	1	3	16	250	0.234	**0.648**	0.623	0.635

Table 4.1: Results from shuffled and un-shuffled data

The results of the experiments are given in Table 4.1. Here, as well as in the upcoming result tables, the best scores will be displayed in **bold**. The numbers show that shuffling affects the recall, precision and F1 score positively and only the precision score declines slightly. As shuffling makes the model more robust for a diverse range of texts structures and sentence sequences, all the succeeding experiments will be continued with shuffled inputs of the datasets. Hence in the following of result tables this column will be dropped due to its redundancy.

The optimal token length for sentences

BERT(je) has a limit in token length per sentence input. The model takes sentences with a maximum of 512 tokens and exceeding tokens are dropped (Lin et al., 2020). To prevent this from happening, a preprocessing script was used with which a maximum token length per sentence could be set for the input. Sentences surpassing this maximum are split—divided into two, with the second sentence proceeding from the exceeding token onwards. To test whether and how a variation in this parameter effects the results, a series of 8 models were run with a range of maximum token lengths. The specific lengths and the results of the inputs are given in Table 4.2.

The overview demonstrates that no sentence surpasses the length of 128 tokens as the results are identical for the lengths of 128 and longer. The optimal results within this set of lengths for loss, precision, recall and F1 are when the maximum is set to 96. While lengths between 96 and 112 or 64 and 96 could be examined for further optimisation of the maximum length, this work will be left for later research. The upcoming experiments are done with a maximum token length of 96 and this column will be omitted in the overview tables henceforth.

[2]the scripts for setting maximum sentence length, shuffling, and conversion to json format were generously provided to me by Sophie Arnoult

max len	seed	epochs	batch size	eval steps	loss	P	R	F1
16	1	3	16	250	0.191	0.657	0.655	0.656
32	1	3	16	250	0.189	0.624	0.649	0.637
64	1	3	16	250	0.179	0.625	0.656	0.640
96	1	3	16	250	**0.177**	**0.645**	**0.673**	**0.659**
112	1	3	16	250	0.179	0.640	0.659	0.649
128	1	3	16	250	0.180	0.645	0.658	0.652
360	1	3	16	250	0.180	0.645	0.658	0.652
512	1	3	16	250	0.180	0.645	0.658	0.652

Table 4.2: Results of a set of experiments to examine the optimal maximum token length per sentence

Random seed

The hyper-parameter of the random seed in BERT has been found to affect weight initialisation and training data order (Dodge et al., 2020). A set of 4 experiments were conducted to examine the effects of adjusting this parameter. For a more exhaustive research, a larger set of experiment will be needed to achieve more definite results. The outcome of the set of runs can be found in Table 3.

While the loss very slightly drops with a random seed of 4 and precision rises mod-

seed	epochs	batch size	eval steps	loss	P	R	F1
1	3	16	250	0.177	0.645	**0.673**	**0.659**
4	3	16	250	**0.176**	0.630	0.652	0.641
8	3	16	250	0.183	0.633	0.630	0.631
16	3	16	250	0.186	**0.651**	0.605	0.627

Table 4.3: Results of examinations with various random seeds

erately with a random seed of 16, the overall results of a random seed of 1 are more preferable. Especially with a preference for higher recall in mind, this outcome is most ideal. The succeeding experiments will therefore be conducted with 1 as a random seed and the column indicating this will be dropped in the following.

The Batch Size

The batch size is the number of sentences in the training data that are given as input per iteration of the neural network. Larger batch-sizes have been found to increase the speed and reduce run time, as frequency of parameter readjustment is lower. Additionally, when the model is learning larger sets at a time, the batch is averaged out over a potentially more diverse range of data. In case of a smaller batch size, the set is less likely to be representative of the entire dataset. However, as the frequency of evaluating the parameters is increased, the model may learn more precisely and is less likely to move off track. In terms of processing resources, it should be noted that the larger number of sentences in the batch size, the higher the amount of RAM required.

An additional benefit of a larger batch size, is a more complete insight of the contex-

tual information that the model can look at per iteration; this will however not apply to our data as the input given consists of shuffled sentences.

epochs	batch size	eval steps	loss	P	R	F1
3	16	250	0.177	0.645	0.673	**0.659**
3	24	250	**0.169**	0.639	0.656	0.647
3	32	250	0.174	**0.653**	0.656	0.654
3	40	250	0.174	0.624	**0.676**	0.650

Table 4.4: Results of examining various batch sizes

While the loss is the lowest for batch size 24, the precision is the highest for batch size 32, and the recall of batch size 40 is the highest, the overall scores of batch size 16 remained more favourable and we therefore continue with this value. The increase in recall was considered too marginal to apply batch size 40. The column of batch size will be left out of succeeding result tables within 4.1.1, with the exception of Table 4.5.

The Batch Size Revisited

At a later stage in the research, this hyper-parameter was revisited with Dataset 4 (Table 3.15). Informed by the trial phase of setting up these experiments, a set of models were run with a smaller batch size. As shown in Table 4.5 and more specifically in the fourth row, these runs resulted into finding a batch size that led to higher results.

While these outcomes may gravitate towards deciding on batch size 13, further in the process in section 4.1.3, experiments were run that compare batch sizes 12 and 13, for the best overall model. Here, batch size 12 (fourth row, Table 4.13) outperformed batch size 13 (last row, Table 4.14). All experiments described in 4.1.3 are conducted with these values for epochs, batch size and eval steps. Due to the lack of consistency in causality of adjusting the hyper-parameter, batch size was evaluated as not conclusive.

epochs	batch size	eval steps	loss	P	R	F1
3	10	250	0.131	0.683	0.683	0.683
3	12	250	**0.121**	0.740	0.722	0.731
3	13	250	0.125	**0.760**	**0.732**	**0.746**
3	14	250	0.127	0.750	0.717	0.733
3	16	250	0.126	0.707	0.710	0.708

Table 4.5: Results of re-examining batch sizes

Epoch

Epochs are the amount of times the neural network sees the data. For this hyper-parameter, first a set of experiments with epochs within the range of 3 to 6 were conducted a shown in Table 4.6. The best results for this dataset was achieved for epoch 5 for all scores.

In following section, experiments are conducted with various datasets and the op-

timal epoch is reevaluated. Initiating these additional experiments was informed by inconsistencies in other hyper-parameters as the batch size.

epochs	eval steps	loss	P	R	F1
3	250	0.177	0.645	0.673	0.659
4	250	0.180	0.637	0.669	0.652
5	250	**0.175**	**0.657**	**0.676**	**0.666**
6	250	0.178	0.631	0.665	0.648

Table 4.6: Results of testing with a range of epochs

Overview of Hyperparameters

I conclude this section with a brief overview of the selected hyperparameter settings to continue the experimental process with:

- Shuffle: Yes; the decision was made to shuffle the data as performance improved with this settings and this would result in a more robust system for diverse datasets.
- Maximum token length: 96; the best results were reached with 96 as the max token length. Lengths between 96 and 112 or 64 and 96 could be examined for potential further optimisation of the maximum length.
- Random seed: 1; this setting gave the best results within the experiment set where it was compared to higher random seeds.
- Batch size: 12; determined after several follow-up experiments that did not show consistent causal connections between batch size and the results, therefore this remains inconclusive.
- Epoch: 3; while in the first set of experiments, 5 epochs gave a better results with Dataset 1, further on (see Table 4.9) for the larger and more diverse Dataset 4, 3 epochs resulted in the overall best results.

4.1.2 Dataset variations

Experimenting with datasets sizes

A series of experiments were conducted with both epoch 3, and 5 and run with various data sizes and diverse compositions of datasets. This was done to examine both whether (1) the relative improvement in the results of Table 4.6 with epoch 5 consistently hold up, and the effects of variations in (2.1) the size of the training data and (2.2) the distribution of the strata over the datasets.

First a set of two experiments with batch sizes 3 and 5, were run with a larger training data (Dataset 2 in Table 3.13). The results are displayed in the penultimate and

last row of Table 4.7. The second row is shown to facilitate the process of comparison with Dataset 1. These results show that a larger training set gives a lower loss and combined with epoch 5 all scores improve.

Dataset	epochs	batch size	eval steps	loss	P	R	F1
1	3	16	250	0.177	0.645	0.673	0.659
2	3	16	250	0.169	0.645	0.667	0.656
2	5	16	250	**0.168**	**0.654**	**0.674**	**0.664**

Table 4.7: Results of testing with a larger training set

As previously mentioned in chapter 3, after noticing that the stratum LtV did not occur in both the test and dev sets of Dataset 2, a new set of datasets was created with a more even distribution of the strata (Dataset 3). In addition, the first and last rounds are more evenly distributed in these datasets. As annotation quality generally improves over time, the redistribution results in more balanced datasets.

The model with a more balanced dataset in terms of rounds and distribution resulted into improvements in all scores, except the loss (see Table 4.8). When combined with batch size 5 only the loss improves slightly, while recall stays the same and precision and F1 drop.

Dataset	epochs	batch size	eval steps	loss	P	R	F1
1	3	16	250	0.177	0.645	0.673	0.659
3	3	16	250	0.138	**0.742**	**0.696**	**0.718**
3	5	16	250	**0.137**	0.727	**0.696**	0.711

Table 4.8: Results of testing with Dataset 3: giving a more even distribution of strata in the datasets

Finally, a set of experiments were performed, where the dataset has a larger training set, in addition to the more even distribution of the rounds and strata. As these gave the highest scores so far, as given in Table 4.9, it was decided that all the succeeding tests will be continued with this dataset and epoch 3. As the differences in results with epoch 5 are marginal and inconsistent, determining the optimal settings for this parameter is inconclusive.

Dataset	epochs	batch size	eval steps	loss	P	R	F1
1	3	16	250	0.177	0.645	0.673	0.659
4	3	16	250	**0.126**	**0.707**	**0.710**	**0.708**
4	5	16	250	0.128	0.706	0.705	0.705

Table 4.9: Results of testing with Dataset 4: giving a more even distribution of strata in the datasets and a larger training set

A systematic overview of dataset variations

In order to create a structured overview of the effects of reducing the size of the training data, a set of models were run where the size of the shuffled training data was systematically adjusted. A python script was used to create the training datasets that are a percentage of the one in Dataset 4. The script continued as long as the indicated percentage was not reached, so the files contain (very) slightly less than the numbers indicated in the first column of Table 4.10 as they are rounded up.

% of the data	loss	P	R	F1
100 %	**0.121**	**0.740**	0.722	0.731
80%	0.125	0.724	**0.740**	**0.732**
60%	0.133	0.710	0.717	0.713
40%	0.142	0.708	0.685	0.696
20%	0.158	0.630	0.653	0.641

Table 4.10: Results of selected declining sizes of training data

The table, furthermore, shows results of the set of runs. The recall and F-score are interestingly highest with 80% of the training data. The loss and precision are the highest with the largest training data. The results for all scores progressively decline from 60% onwards with each descending step. As an even distribution of strata and annotation rounds can not be ensured for the highest performing dataset (80%), experiments are continued with 100% of Dataset 4.

4.1.3 Label Merges

In the final set of experiments, selected sets of labels were merged with the objective to simplify classification for the algorithm and potentially improve the results. In selecting the labels and merges, the aspects of similarity, interchangeability, and/or whether labels were confusing during annotation were taken into account.

Numeric label merges

The labels for the mainly numeric entities–BANKNR, KENT, PHONE–were individually and collectively merged with the CODE label as shown in Table 4.11. These however resulted into only marginal improvements, with the merging of only the PHONE label resulting in progress for all scores and the BANKNR merge resulting in a rise in the F1 and recall scores. The results were however not sufficient to sacrifice the contextual specificity of preserving these labels and no merges with numeric labels were implemented for the final model.

Combinable label merges

A few labels had the possibility to be grouped under another label due to similarities in these categories. Three merges were implemented and models were run to examine whether this would lead to improved scores. EDU was merged with the ORG label which it could be categorised under without too much contextual loss. This however resulted in a decline of all the scores. LOCderiv was merged with LOC with a similar

merge(s) applied	loss	P	R	F1
none	0.121	0.740	0.722	0.731
BANKNR>CODE	0.122	0.736	0.732	0.734
KENT>CODE	0.124	0.734	0.723	0.728
PHONE>CODE	**0.121**	**0.744**	**0.742**	**0.743**
BANKNR+KENT+PHONE>CODE	0.126	0.728	0.702	0.714

Table 4.11: Results of numeric label merges

aim as for the previous merge, namely to simplify the classification task. The merge resulted in an improved recall and F1 score. USER was merged with the PER label and resulted in a very slight improvement in recall. The results are displayed in Table 4.12. The merges were not applied as their benefits did not outweigh the loss of context when preserving the labels.

merge(s) applied	loss	P	R	F1
none	0.121	**0.740**	0.722	0.731
EDU>ORG	0.123	0.728	0.728	0.728
LOCderiv>LOC	0.121	0.735	**0.737**	**0.736**
USER>PER	**0.119**	0.728	0.726	0.727

Table 4.12: Results of combinable label merges

Merges of labels not requiring anonymisation

The entities with the labels MISC and PJ, will remain un-anonymised. The labels were created in order to bring about more consistency for the classification-task. Both labels were used to label tokens that stand-out form-wise and are similar in shape to the other NEs. These labels were additionally considered to be the most confusing ones during the annotation process. Models were run where the labels were simultaneously and individually left out of the classification task and replaced with O, the outside label. Leaving out only the MISC label resulted in a slight improvement for the recall and F1 score and a drop in the loss. Replacing only the PJ with an O, led to improvements in all scores. Leaving out both labels resulted in the highest scores for the P, R and F1 scores, however the loss rose. These results made the model where PJ and MISC were dropped the best model so far.

merge(s) applied	loss	P	R	F1
none	0.121	0.740	0.722	0.731
MISC>O	0.112	0.735	0.732	0.734
PJ+MISC>O	0.195	**0.782**	**0.766**	**0.774**
PJ>MISC	0.118	0.751	0.742	0.746
PJ>O	**0.105**	0.764	0.763	0.764

Table 4.13: Results selected label merges

Miscellaneous merges

Additional models that were run to test combinations of merges and parameters settings that previously led to improved results, as well as a test on the effect of binary classification, are described in the following.

A model was run where two merges that independently had given positive results were combined: PHONE was merged with CODE, and both PJ and MISC were replaced with O. As shown in the third row of Table 4.14, this led to a higher recall than that of the previously best model (see Table 4.13). Additionally, this shows improvements in all scores, except the loss, in comparison to the results of the run without any merges. As the label PHONE carries context-specificity that is more valuable than the model's marginal improvement in recall, in comparison to the previous test, this merge was not applied.

To examine the results of binary classification, the labels PJ and MISC were replaced with O and all the other labels—referred to as REST in the table—were replaced with a superordinate label ANNO, as these entities would be anonymised. The results, given in the penultimate row of Table 4.14, show improvements for all scores except the loss. In cases where context preservation is not required, binarization would be most advantageous.

merge(s) applied	loss	P	R	F1
none	**0.121**	0.740	0.722	0.731
PJ+MISC>O, PHONE>CODE	0.194	0.774	0.767	0.769
(binary) PJ+MISC>O, REST>ANNO	0.172	**0.807**	**0.799**	**0.803**
(batch size: 13) PJ+MISC>O	0.190	0.755	0.729	0.742

Table 4.14: Results selected label merges

The last row, shows the results of a run where PJ and MISC are merged with O and the batch size is set to 13, as these adjustments have shown to improve the scores in previous tests. Combining these adjustments surprisingly did not lead to improved results in comparison with the results of the run with only the PJ and MISC dropped and batch size 12. This induces question about the consistency in which a specific batch size effects the results. As indicated in section 4.1.1, batch size tests were determined to be inconclusive.

For the final model batch size was kept at 12 and the only merge applied was that of the labels PJ and MISC with O.

4.2 Reflection

In this chapter, the BERTje-based NERC model for context-preserving anonymisation was chosen through three experimental phases: hyperparameter settings, datasets experiments and label merges.

We started with a search for the optimal hyperparameter settings, where almost all model were runs with Dataset 1 (Table 3.12). First, a shuffled dataset was found to be the most robust and suitable for our diverse data with ten strata. In future work, if strata specific models are built, it is worthwhile to experiment with un-shuffled data. As some strata have a very regular structure in text form and sentence sequences, perserving this in fine-tuning may effect performance positively. The optimal maximum

token length per sentence was found to be 96 within the set of experiments conducted. In further research lengths between 64 and 96 and 96 and 112 may be worthwhile to examine. The random seed of 1 was found to give the best performance, albeit within a small set of runs. With a larger range in random seeds in the experiment set, result could be improved. A total of 10 models were run to examine the best scores for batch size. These were achieved with Dataset 4 (Table 3.15) and the merges PJ and MISC with O implemented and batch size 12. Epoch 3 emerged as the best option, after a series of experiment sets with a variety of datasets. As the last two hyper-parameters have been rather inconsistent, their effects are assessed as inconclusive within these sets of experiments. For the cases of batch size and epoch, follow-up experiments were done at a later time, with different datasets and/or merges applied. While the course of experiments could be conducted more systematically with more time available or in future work, doing these follow-up experiments led to better results and made the inconsistency of these parameters visible.

In the second phase of experiments, dataset variations in terms of size, distribution of annotation rounds, and strata were tested. The highest recall was achieved with Dataset 4, with a larger training data and an even distribution of strata and annotation rounds for the train, test, and dev sets. All three adjustments applied had positive effects on the results. Subsequently, a systematic set of 5 experiments were conducted with 100, 80, 60 40, and 20 percent of the training data of Dataset 4, to inspect how the size of the training set effects the results in a structured way. Overall a higher dataset led to higher scores, yet the recall and F1 were highest with 80% of the data. As an even distribution of strata can not be assured for the latter, the experiments were continued with the full version of Dataset 4. In future work, systematical tests of datasets sizes with even distribution in strata and annotation rounds could be done to uncover whether there is an upper-limit in the size of training data to improve recall.

Finally, a series of models were runs, where label merges were applied for (1) a select numeric label set, (2) combinable labels, (3) labels not requiring anonymisation, and (4) miscellaneous merges: binarization, and combinations of merges and parameters settings that previously led to improved results. The merges with numeric and combinable labels rarely led to improvements. In cases of improved scores, the resulting trade-off with preserving contextual information was not preferable. For the labels not requiring anonymisation, PJ, and MISC, omitting both gave the best results. The binary anonymisation model received the highest overall score and is most suitable when preservation of context is not required.

The final model with the highest scores for context-preserving anonymisation resulted from the hyperparameters indicated at the end of section 4.1.1 with Dataset 4 (Table 3.15) and with the labels PJ and MISC dropped, shown in the third row of Table 4.13[3].

[3] An evaluation matrix of all the labels of the final model, useful for future work, is given in Appendix E.

Chapter 5

Conclusions and Recommendations

5.1 Conclusions

In this project a Dutch NERC system for police records based on BERTje—a pre-trained transformer-based model—was created through the steps described in the following.

First, the choice for a BERT-based system was made, as fine-tuning language models are the current state of the art for NER. Additionally, they perform well with relatively small datasets, making the model more suitable for this project. Among other multi- and monolingual Dutch BERT models, BERTje was chosen on the basis of two qualities of the pre-training corpora: diversity and magnitude. While comparisons with, in some cases slightly higher, research results of other Dutch BERT models were taken into account, this remains an inconclusive ground to decide on a model as the fine-tuning data for these works varies considerably from the dataset used for this project.

Subsequently, an annotation process was carried out to create a NERC corpus for Dutch police reports. Here, the raw dataset was first filtered to distill a diverse natural language dataset with a set of ten strata. Two samples were drawn for the first round of annotations and the IAA, and the second round. Annotation guidelines were created to guide a team of four annotators, who annotated a total of 14439 NEs in 950 police reports and achieved near perfect IAA scores. The resulting dataset was distributed into four datasets of various sizes and distribution in strata and annotation rounds.

Finally, the pre-trained BERTje was fine-tuned through adjusting hyper-parameters, testing with the four datasets and a structured testing round with training data-size, and merging specific labels. As a result, one model was chosen as the optimal system for NERC-based context-preserving anonymisation of police records. While the conversion from NERC labels to anonymising the text still remains to be done, this merely requires relatively minor post-processing steps.

Through these steps, answers were found to the research questions formulated to initiate this project:

1. (How) can (an) NERC-based anonymisation tool(s) be applied effectively to the domain of police reports?

2. Is it possible to create one anonymisation tool for a variety of writing style and

text structures?

A NERC-model for context-preserving anonymisation for police reports was created through fine-tuning a BERTje model with Dataset 4 (Table 3.15), with a diverse distribution of the two annotation rounds and ten strata. The dataset with has a relatively larger training set, and non-anonymising labels that were introduced merely to facilitate the model were removed. It is an anonymisation tool based on a NER classifier with a total of 14 labels. The final hyper-parameter settings (section 4.1.1), and label merges were applied with the chosen dataset as a result of 56 experiments.

While the binary model achieved the highest precision, recall and F1 scores, in selecting the suitable model, its deficiency—the fact that it is not context preserving, will unfortunately outweigh these positive results. A context-preserving quality is desired for the anonymisation tool for this project.

The final model proves that it is additionally possible to create one anonymisation tool for a variety of writing styles and text forms as the ten strata were created based on their diversity in term of these aspects.

5.2 Recommendation

In the further development of this project, various steps of the process can be refined and improved.

In preprocessing the data, the most frequently occurring texts without NEs that require anonymisation were filtered out as many of them were standard texts that occurred over 10000 times. As a result, the datasets used to build the models are focused on documents containing NEs that require anonymisation. To get an idea of how the system performs on unfiltered data, it is worth experimenting with models where these excluded documents are added back to the datasets.

In the case that more data are annotated for the training set, it is advisable to leave out the MISC and PJ labels as words given these labels did not have the negative effect on the result that was initially anticipated.

An important insight gained from creating the input data in this project is the positive effect of ensuring a diverse and even distribution of all strata and annotation rounds over the train, test and development sets. In creating the datasets in the further development of this project, it is important to apply the lessons learned.

In fine-tuning, it could be useful to experiment with maximum tokens lengths per sentence between 96 and 112 and perhaps those between 64 and 96, to further optimise results. Additionally, trying out models that are strata specific may improve results, as an even distribution of LtV in the datasets led to an improvement in the results. Especially for some strata that have a very regular text formats, models could be fine-tuned with a smaller dataset and still reach satisfying results and simultaneously optimise efficiency by decreasing the amount of labour intensive annotation work. In addition, as texts structures are more similar within a stratum, the step of shuffling the sentences could be left out to examine whether preservation of sentence order—giving a more consistent contextual information—may improve performance.

In terms of merges, in the continuation of the work at CBS on cybercrime categorisation of police records, it is recommended to additionally experiment to run the model without the #WEBAPP label. The experiments done in this work have included the label. In the output of these models, entities with the #WEBAPP label can be used as

a cue for cybercrime categorisation. Moreover, in both cases, it can be tested whether the mis-association patterns of certain NE's with the cybercrime category still occurs in these set-ups as mentioned in the introduction.

5.3 Future work

Various other approaches to the problem of NERC-based anonymisation can be experimented with in future work.

Models that are definitely worth trying are RobBERT (Delobelle et al., 2020), as their corpus could potentially be more aligned with the police reports, as well as mBERT as applied by Wu and Dredze (2019). While the latter reached state-of-the-art results, it may not excel for the police reports dataset, as it was pre-trained with wikipedia texts and these scores resulted from fine-tuning with news documents. As RobBERT was pre-trained with a massive corpus of over 39 GB crawled from the internet that is more likely to contain texts with non-standard language, it may lead to improvements in results. Alternatively, if time and resources allow, CBS could pre-trained their own BERT or RobBERT model with a police reports corpus, as well as other freely available Dutch corpora consisting of informal natural language. In this case, it is possible to experiment with various combinations of pre-training steps pre-training steps that been found to be more effective. For instance, by combining two alternative pre-training steps such as giving multiple sentences as input in the MLM step as in the RoBERTa models (Liu et al., 2019; Delobelle et al., 2020), as well as replacing NSP with SOP (Lan et al., 2020).

It is useful to try previously state-of-the-art models, as results may vary given the input and this domain has not been tested in previous works. Rule and knowledge-based models may be effective for specific NEs that are regular in form such as various numeric and (digital) contact information (e.g. phone numbers, postal code, email addresses etc.). In addition, the task of detecting companies, physical addresses and locations can be successfully applied with a lexicon-based model[1]. However rule- and knowledge based systems are not always as effective. Names of citizens that are not standard in the Dutch context and may not be included in these lexicons. Such methods may have grave consequences as the identities of citizen whose names do not occur in these lexicons will not be anonymised and protected. This could result in refraining from filling police reports. Relying on models (pre-)trained on the sentence and document structure may be a safer bet for these NEs, rather than knowledge-based systems that are always limited.

Testing various rule- and knowledge-based systems, as well as ML approaches to find out which work best for which NEs and combining the optimal models could result in a high-quality classifier for Dutch police reports.

[1]Databases for various NEs are available, e.g. Basisregistratie Adressen en Gebouwen (BAG): `https://bag.basisregistraties.overheid.nl`, De Nederlandse Voornamenbank: `https://www.meertens.knaw.nl/nvb/`

Bibliography

E. Alfonseca and S. Manandhar. An Unsupervised Method for General Named Entity Recognition and Automated Concept Discovery. 2002.

M. Asahara and Y. Matsumoto. Japanese Named Entity extraction with redundant morphological analysis. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - NAACL '03*, volume 1, pages 8–15, Edmonton, Canada, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073447. URL http://portal.acm.org/citation.cfm?doid=1073445.1073447.

D. Benikova, C. Biemann, and M. Reznicek. NoSta-D named entity annotation for German: Guidelines and dataset. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2524–2531, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/276_Paper.pdf.

D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel. Nymble: a high-performance learning name-finder. In *Proceedings of the fifth conference on Applied natural language processing* -, pages 194–201, Washington, DC, 1997. Association for Computational Linguistics. doi: 10.3115/974557.974586. URL http://portal.acm.org/citation.cfm?doid=974557.974586.

J. Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, Apr. 1960. ISSN 0013-1644, 1552-3888. doi: 10.1177/001316446002000104. URL http://journals.sagepub.com/doi/10.1177/001316446002000104.

M. Collins and Y. Singer. Unsupervised Models for Named Entity Classification. page 11, 1999.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural Language Processing (Almost) from Scratch. *NATURAL LANGUAGE PROCESSING*, page 45, 2011.

W. de Vries, A. van Cranenburgh, A. Bisazza, T. Caselli, G. van Noord, and M. Nissim. BERTje: A Dutch BERT Model. *arXiv:1912.09582 [cs]*, Dec. 2019. URL http://arxiv.org/abs/1912.09582. arXiv: 1912.09582.

P. Delobelle, T. Winters, and B. Berendt. RobBERT: a Dutch RoBERTa-based Language Model. *arXiv:2001.06286 [cs]*, Sept. 2020. URL http://arxiv.org/abs/2001.06286. arXiv: 2001.06286.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019. URL http://arxiv.org/abs/1810.04805. arXiv: 1810.04805.

J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, and N. Smith. Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping. *arXiv:2002.06305 [cs]*, Feb. 2020. URL http://arxiv.org/abs/2002.06305. arXiv: 2002.06305.

A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, pages 2047–2052, Montreal, Que., Canada, 2005. IEEE. ISBN 978-0-7803-9048-5. doi: 10.1109/IJCNN.2005.1556215. URL http://ieeexplore.ieee.org/document/1556215/.

R. Grishman and B. Sundheim. Message Understanding Conference- 6: A Brief History. page 6, 1996.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9: 1735–1780, 1997.

B. Kleinberg, M. Mozes, Y. van der Toolen, and B. Verschuere. Netanos - named entity-based text anonymization for open science, Jun 2017. URL osf.io/w9nhb.

J.-C. Klie, M. Bugert, B. Boullosa, R. E. de Castilho, and I. Gurevych. The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9. Association for Computational Linguistics, June 2018. URL http://tubiblio.ulb.tu-darmstadt.de/106270/.

G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer. Neural Architectures for Named Entity Recognition. *arXiv:1603.01360 [cs]*, Apr. 2016. URL http://arxiv.org/abs/1603.01360. arXiv: 1603.01360.

Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv:1909.11942 [cs]*, Feb. 2020. URL http://arxiv.org/abs/1909.11942. arXiv: 1909.11942.

J. Lin, R. Nogueira, and A. Yates. Pretrained Transformers for Text Ranking: BERT and Beyond. *arXiv:2010.06467 [cs]*, Oct. 2020. URL http://arxiv.org/abs/2010.06467. arXiv: 2010.06467.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*, July 2019. URL http://arxiv.org/abs/1907.11692. arXiv: 1907.11692.

A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003* -, volume 4, pages 188–191, Edmonton, Canada, 2003. Association for Computational Linguistics. doi: 10.3115/1119176.1119206. URL http://portal.acm.org/citation.cfm?doid=1119176.1119206.

B. Medlock. An introduction to NLP-based textual anonymisation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, May 2006. European Language Resources Association (ELRA). URL `http://www.lrec-conf.org/proceedings/lrec2006/pdf/200_pdf.pdf`.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*, Sept. 2013. URL `http://arxiv.org/abs/1301.3781`. arXiv: 1301.3781.

R. Motwani and S. U. Nabar. Anonymizing Unstructured Data. *arXiv:0810.5582 [cs]*, Nov. 2008. URL `http://arxiv.org/abs/0810.5582`. arXiv: 0810.5582.

D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticæ Investigationes*, 30(1):3–26, 2007. ISSN 0378-4169. doi: https://doi.org/10.1075/li.30.1.03nad. URL `https://www.jbe-platform.com/content/journals/10.1075/li.30.1.03nad`. Publisher: John Benjamins Type: Journal Article.

I. Neamatullah, M. M. Douglass, L.-w. H. Lehman, A. Reisner, M. Villarroel, W. J. Long, P. Szolovits, G. B. Moody, R. G. Mark, and G. D. Clifford. Automated de-identification of free-text medical records. *BMC Med Inform Decis Mak*, 8(1):32, Dec. 2008. ISSN 1472-6947. doi: 10.1186/1472-6947-8-32. URL `http://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/1472-6947-8-32`.

N. Oostdijk, M. Reynaert, V. Hoste, and I. Schuurman. The Construction of a 500-Million-Word Reference Corpus of Contemporary Written Dutch. In P. Spyns and J. Odijk, editors, *Essential Speech and Language Technology for Dutch: Results by the STEVIN programme*, pages 219–247. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-30910-6. doi: 10.1007/978-3-642-30910-6_13. URL `https://doi.org/10.1007/978-3-642-30910-6_13`.

R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, page III–1310–III–1318. JMLR.org, 2013.

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *arXiv:1802.05365 [cs]*, Mar. 2018. URL `http://arxiv.org/abs/1802.05365`. arXiv: 1802.05365.

L. Plamondon, G. Lapalme, and F. Pelletier. Anonymisation de décisions de justice. page 10, 2020.

T. Poibeau and L. Kosseim. Proper Name Extraction from Non-Journalistic Texts. 2001.

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving Language Understanding by Generative Pre-Training. page 12, 2018.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language Models are Unsupervised Multitask Learners. page 24, 2019.

L. Rau. Extracting company names from text. In *Proceedings The Seventh IEEE Conference on Artificial Intelligence Application*, pages 29,30,31,32, Los Alamitos, CA, USA, feb 1991. IEEE Computer Society. doi: 10.1109/CAIA.1991.120841. URL https://doi.ieeecomputersociety.org/10.1109/CAIA.1991.120841.

E. F. T. K. Sang. Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. *arXiv:cs/0209010*, Sept. 2002. URL http://arxiv.org/abs/cs/0209010. arXiv: cs/0209010.

Y. Shao, C. Hardmeier, and J. Nivre. Multilingual Named Entity Recognition using Hybrid Neural Networks. page 4, 2016.

C. Sun, X. Qiu, Y. Xu, and X. Huang. How to Fine-Tune BERT for Text Classification? *arXiv:1905.05583 [cs]*, Feb. 2020. URL http://arxiv.org/abs/1905.05583. arXiv: 1905.05583.

P. J. O. Suárez, B. Sagot, and L. Romary. Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. page 9, 2019.

L. Sweeney. Replacing Personally-Identifying Information in Medical Records, the Scrub System. page 5. URL 1996.

Y. Tsuruoka and J. Tsujii. Boosting precision and recall of dictionary-based protein name recognition. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine* -, volume 13, pages 41–48, Sapporo, Japan, 2003. Association for Computational Linguistics. doi: 10.3115/1118958.1118964. URL http://portal.acm.org/citation.cfm?doid=1118958.1118964.

M. van der Sangen. Europese privacywet: het CBS is er klaar voor, 2018. URL https://www.cbs.nl/nl-nl/corporate/2018/21/europese-privacywet-het-cbs-is-er-klaar-voor.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin. Attention is All you Need. 2017.

H. Vico and D. Calegari. Software Architecture for Document Anonymization. *Electronic Notes in Theoretical Computer Science*, 314:83–100, June 2015. ISSN 15710661. doi: 10.1016/j.entcs.2015.05.006. URL https://linkinghub.elsevier.com/retrieve/pii/S1571066115000298.

S. Wu and M. Dredze. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1077. URL https://www.aclweb.org/anthology/D19-1077.

V. Yadav and S. Bethard. A Survey on Recent Advances in Named Entity Recognition from Deep Learning models. *arXiv:1910.11470 [cs]*, Oct. 2019. URL http://arxiv.org/abs/1910.11470. arXiv: 1910.11470.

Appendix A

Artificial Police Reports

In the following, artificial examples of texts per category or strata are given. As the form of texts vary per stratum and sometimes even within a stratum. See table A.1 for a quick overview of specific forms per stratum.

	Toelichting	Verklaring
LMIO	NL: police writing *or* NL: Similar recurring text with name of police employee leading case and unique reference number + Standard text on LMIO	NL: citizen writing *or* NNL: form + NL: Beschrijving part of form: citizen writing
non-LMIO	NNL: form-like mostly caps + NL: police writing	NL: citizen writing
police	NNL: Mostly caps + NL: police writing	-

Table A.1: Text format per source label and section

Of the various forms mentioned in this table, examples of the following strata are given below:

1. LMIO Toelichting: police report
2. LMIO Verklaring: citizen report
3. LMIO Verklaring: filled-in form
4. non-LMIO Toelichting: standard text
5. non-LMIO & police Toelichting: police report
6. non-LMIO Verklaring

In order to avoid redundancy and maintain simplicity, some examples have been left out of merged with another category.

A.1 LMIO Toelichting

Pluk van de Pettenflat
18.02.2019

==========

LMIO (Landelijke Meldpunt Internet Oplichting) gecontacteerd. Rekeningnummer komt 54 keer voor met oplichting. Op dit moment wordt de zaak niet opgepakt.

Aangifte Facebookoplichting opgenomen.
Puck van de Pettenflat

A.2 LMIO Verklaring: citizen report

Ik ben opgelicht via Facebook en wil hier aangifte voor doen. Op zaterdag 7 februari, zag ik een advertentie in de Facebook groep "zwementhousiasten" voor een opblaas flamingo voor 15euro van dhr. Rientje Pannenkoek en heb hem een bericht gestuurd via messenger. Vervolgens kreeg ik het verzoek mijn adresgegevens op te sturen en het bedrag over te maken naar NL12SNSB436327563285 over te maken. Zou het binnen 2 werkdagen ontvangen. Heb dit direct gedaan en gewacht maar had het 5 dagen later nog niet binnen. Heb dhr. Pannekoek meerdere berichten gestuurd, maar heb geen bericht meer gekregen. Deze man is een oplichter en hij moet worden gestopt. Het gaat mij niet het geld, maar de principe. Ik had gehoopt er veel zwemplezier mee te hebben, dus de teleurstelling is groot.

A.3 LMIO Verklaring: filled-in form

Via www.politie.nl is aangifte gedaan ter zake oplichting door de in deze registratie genoemde aangever ten aanzien van een wederpartij.
Er is een product besteld via een online handelsite (bv Marktplaats of Speurders), dan wel via een malafide webwinkel of via social media.
Er heeft een betaling plaatsgevonden in de veronderstelling dat de aangever het gekochte product zou ontvangen. Ondanks een betaling, ter verkrijging van het gekochte/afgesproken beloofde product, werd het toegezegde niet verstrekt/verzonden/geleverd/ontvangen.

De informatie in deze aangifte wordt verwerkt door het Landelijk Meldpunt Internet Oplichting (LMIO), onderdeel uitmakend van het Landelijk Service Centrum eCrime (LSCeC). Het LMIO is ontstaan uit een samenwerkingsovereenkomst met het Arrondissementsparket Haarlem en de politie en verzorgt de landelijke intake, analyse, coödinatie van internetgerelateerde oplichting.
Aan de hand van criteria start het LMIO een opsporingsonderzoek en verwerkt dit in een onderzoeksdossier.
Voorbereide onderzoeken worden door het LMIO toegezonden aan de politie & het regioparket binnen wiens eenheid en arrondissement een verdachte woonachtig is met het verzoek het dossier verder af te handelen.

Voor meer informatie neem contact op met het Landelijk Meldpunt Internet Oplichting te bereiken via email: lmio@polite.nl of via telefoonnummer 011-2345678 (bereikbaar van maandag tot en met vrijdag tussen 08.00 en 17.00 uur - alleen voor politie).

Referentie.
Wederpartij Voorna(a)m(en) Max
Wederpartij Tussenvoegsel(s) van der
Wederpartij Achternaam Kwast
Wederpartij Straatnaam Torenstraat
Wederpartij Huisnummer 565
Wederpartij Woonplaats Egwijk
Wederpartij Land Nederland
Wederpartij Postcode 2356 78
Wederpartij E-mailadres wederpartij maxkwast@facebook.com
Wederpartij Telefoonnummer 072-5453480
Wederpartij Mobiele telefoonnummer +31658307548
Conflict Betreft het een handelssite? Geen handelssite, maar marketplace
Conflict Hoe bent u hier terechtgekomen? Overigen
Conflict De webshop facebook marketplace
Conflict Uw gebruikersnaam handelssite Pluk Pet
Conflict Gebruikersnaam wederpartij Max van der Kwast
Conflict Omschrijving conflict opblaas flamingo besteld maar kwam nooit aan, had helaas al 25eu betaald!
Conflict Uw bankrekeningnummer NL12SNSB54654675657
Conflict Datum betaling 07-02-2020
Conflict Bedrag aankoop 15euro
Conflict Betalingsmethode Buitenlandse overschrijving (geen IBAN)
Conflict Bankrekeningnummer wederpartij NL12SNSB645873653630
Conflict Naam rekeninghouder wederpartij M van der Kwast
Conflict Verstrekken e-mailadres Ja
Overzicht Zijn de gegevens correct en naar waarheid ingevuld? Ja

A.4 non-LMIO Verklaring: standard text

AANGIFTE OPGENOMEN MIDDELS INTERNET

A.5 non-LMIO & police Toelichting: police report

Melding over aanreden lantarenpaal ontvangen. Ter plaatse gegaan. Lantarenpaal bij pannenfabriek bleek een flink deuk te hebben opgelopen en staat onstabiel.Melder werkt bij fabriek, heeft bestuurder en voertuig/kenteken nr niet gezien.

Plaats: DEN HELDER
Naam melder: MOLENAAR
INCIDENT: 478327

MELDER: 064343243
12-04-2018

A.6 non-LMIO Verklaring: citizen report

Ik doe aangifte ivm een gestolen object. Het object is een opblaas flamingo. Het lag in de tuin terwijl wij een weekend weg waren. Het is roze van kleur en is ongeveer een meter lang en kan gebruikt worden om te drijven in het water. Het is gestolen aan de Lakenstraat 45 te Wageningen, ons adres. Het moet op de nacht van zaterdag 5 april en zondag 6 april zijn gebeurd. De buurman had het die zaterdagavond nog gezien. Zondagochtend viel het hem op dat het weg was en heeft een melding gemaakt in de buurtsWhatsapp-groep. Wij maken lazen dit pas in de middag en hebben onze dochter gevraagd te gaan kijken, die het ontbreken van de flamingo bevestigde. Ik voel mij niet meer veilig in mijn straat. Wij wonen al 40 jaar hier en zoiets naars is ons nog nooit overkomen. De bewoners van het huis zijn, ikzelf en mijn vrouw: Jan Klaasen en Katrijn Pieters-Klaasen. Wij hopen dat de flamingo gauw terug wordt gevonden. Onze telefoon nummers zijn: +31678904567, +31657892311, 011-2378530, uw kunt ons altijd bellen, als er meldingen binnen komen.

Appendix B

NE Annotatie Handleiding: *voor het anonimiseren van processen verbaal voorgaande aan CBS cybercrime categorisatie*

Named Entity Recognition(NER) is de taak om eigennamen of named entities (NE) in teksten te herkennen. Het proces wordt afgelegd in twee stappen en vereist:

1. Named Entity Detection (NED): het detecteren van een woord(frases) of zinsdelen die onder een eigennaam vallen;

2. Named Entity Classification (NEC): het kiezen van een semantische categorieën voor deze benoemde entiteiten.

De NE semantische klassen voor deze taak zijn:

1. #PER: eigennamen die personen aanduiden
2. #USER: gebruikersnamen op virtuele platforms
3. #ORG: eigennamen van organisaties, bedrijven of instellingen
4. #EDU: onderwijsinstelling
5. #WEBAPP: virtueel platform
6. #LOC: plaatsen beschreven met een eigennaam (inclusief adressen)
7. #LOCderiv: plaats derivatie
8. #DATE: zinsdelen die (geboorte)data aangeven, eigennamen voor tijdseenheden
9. #MAIL: emailadres

10. #PHONE: telefoonnummer
11. #BANKNR: (numerieke) bankrekeninggegevens
12. #KENT: kentekennummers
13. #PJ: politie jargon
14. #MISC: woorden die lijken op NEs, maar niet geanonimiseerd hoeven te worden
15. #CODE: numerieke codes die niet onder de bovenstaande klassen vallen
16. #OTH: niet-numerieke woorden en zinsdelen die niet onder de bovenstaande klassen vallen, maar geanonimiseerd zouden moeten worden

NEderiv: afgeleid van eigennamen

[Noord Duitse]$^{\#LOCderiv}$ stammen
De zorgen voor een zeespiegelstijging is heerst onder [Rijnmondse]$^{\#LOCderiv}$ gemeentes.
[Tibentaanse]$^{\#LOCderiv}$ filosofie

Hoe vind ik een Named Entity?

1. Named Entities zijn altijd volledige nominale frases. Ze behoren tot zelfstandig naamwoorden.

2. Named Entities zijn aanduidingen voor unieke eenheden die niet door gemeenschappelijke kenmerken worden beschreven.

   [De Kerstman] heeft veel [rendieren], waaronder [Rudolf].

   In het bovenstaande voorbeeld zijn drie nominale frases. *De Kerstman* en *Rudolf* zijn eigennamen. Ondanks dat er misschien meer dan één Rudolf zou kunnen zijn, wijst dat er niet op dat ze meer dan deze naam gemeen hebben. In het geval van "rendieren" zijn er veel gemeenschappelijke kenmerken onder rendieren.

3. Lidwoorden zijn geen deel van de naam.

   De [Kerstman]$^{\#NE}$ is blij.

4. Een eigennaam kan meer dan een woord bevatten.

   [The Lord of the Rings]$^{\#OTH}$ is succesvol verfilmd.
   Het verhaal werd geschreven door [Sophie Redmond]$^{\#PERSON}$.

5. Eigennamen kunnen in elkaar genesteld zijn.

   Ik kocht een t-shirt van [AFC Ajax [Amsterdam]$^{\#LOC}$]$^{\#ORG}$.

[Zondag met [Lubach]$^{\#PER}$]$^{\#OTH}$ is soms best vermakelijk.

6. Titels, begroetingen en eigenaren behoren niet tot een complexe eigennaam. Eigenaren kunnen op zichzelf ook eigennamen zijn.

   We luisteren naar [Lizzo's]$^{\#PER}$[Coconut oil]$^{\#OTH}$.
   $Meneer[Jansen]^{\#PER}$ vergat zijn hond in het park.

7. Eigennamen kunnen voorkomen als onderdeel van een complexe zelfstandig naamwoord. Hier wordt het gehele woord dat een eigennaam bevat geannoteerd.

   De verkoper stuurde mij een [whatsapp-berichtje]$^{\#WEBAPP}$.
   In een [facebookoproep]$^{\#WEBAPP}$ zag de fiets voorbij komen.
   Zonder [USB-C poort]$^{\#OTH}$ kon de mobiel niet worden verbonden.

8. Wanneer een nominale phrase niet contextueel als een eigennamen of appellatief kan worden bepaald, wordt het niet gemarkeerd als NE.

   De stadspoort is een populaire ontmoetingsplaats.

9. Wanneer een naam de naam is geworden van bepaald voorwerp in de taal en in gebruik niet meer functioneert als een NE wordt het niet geannoteerd.

   [Luna]$^{\#PER}$ hervond de teddybeer achter de bank.
   Voor haar verjaardag kreeg [Ritu]$^{\#PER}$ een bullet journal van haar moeder.

10. In het geval van tellingen met koppeltekens of het uitstellen van een deel van het NE naar latere woorden, wordt het NE-deel geannoteerd alsof het volledig is uitgeschreven.

    De [Eerste]$^{\#OTH}$ en [Tweede Kamer]$^{\#OTH}$ vergaderen tijdelijk digitaal.
    De provincies [Noord-]$^{\#LOC}$ en [Zuid-Holland]$^{\#LOC}$ werden opgesplitst in [1840]$^{\#DATE}$.

Tot welke semantische klasse behoort een eigennaam?

Zie de semantische klasse tabellen per label voor verduidelijking.

- NE met spelfouten, moet ook worden geannoteerd.
- Jaartallen in #ORG worden niet gemarkeerd:

  [Janelle Monáe]$^{\#PER}$ trad op op [North Sea Jazz Festival]$^{\#ORG}$ [2019]$^{\#DATE}$.

- Als een NE in een token voorkomt met een ander woord(deel), de hele token labelen:

[OCP/J. Slingers]$^{\#PER}$
[marktplaats/emailaccount]$^{\#WEBAPP}$
Hij heeft toegang tot mijn [marktplaatsaccount]$^{\#WEBAPP}$

Appendix C

Named Entity Annotation Guidelines
In the anonymisation of police reports preceding CBS' cybercrime categorisation task

Named Entity Recognition(NER) is the task of recognising proper names or named entities(NE) in texts. It is a two step process and requires:

1. Named Entity Detection (NED): detecting the tokens that belongs to a named entity;
2. Named Entity Classification (NEC) - assigning these named entities to semantic categories.

The NE semantic classes for this task are:

1. #PER: proper names denoting persons
2. #USER: virtual platform usernames
3. #ORG: proper names of organisations, companies or institutions
4. #EDU: educational institution
5. #WEBAPP: virtual platform
6. #LOC: places described with a proper name (addresses included)
7. #LOCderiv: derivation of places described with a proper names
8. #DATE: phrases indicating dates (of birth), time units exceeding hours
9. #MAIL: email address
10. #PHONE: phone number

11. #BANKNR: mainly numerical bank account details
12. #KENT: license plate number
13. #PJ: police jargon
14. #MISC: words have the form of NEs, but do not require anonymisation
15. #CODE: numeric codes that are not contained in the classes above
16. #OTH: NEs that require anonymisation and are not contained in the classes above

NEderiv: appellatives derived from proper names

[Noord Duitse]$^{\#LOCderiv}$ stammen
De zorgen voor een zeespiegelstijging is heerst onder [Rijnmondse]$^{\#LOCderiv}$ gemeentes.
[Tibentaanse]$^{\#LOCderiv}$ filosofie

How do I find a Named Entity?

1. Named Entities are always full nominal phrases. They belong to the category noun.

2. Named Entities are designations for unique units which are not described by common characteristics.

   [De Kerstman] heeft veel [rendieren], waaronder [Rudolf].

   There are three nominal phrases in the example above.*De Kerstman* and *Rudolf* are proper names as they may be more than one Rudolf, but their commonality would be the name. While in the case of "rendieren", there are many common characteristics among reindeers.

3. Determiners are not a part of the name.

   De [Kerstman]$^{\#NE}$ is blij.

4. Proper names can contain more than one token.

   [The Lord of the Rings]$^{\#OTH}$ is succesvol verfilmd.
   Het verhaal werd geschreven door [Sophie Redmond]$^{\#PERSON}$.

5. Proper names can be nested.

   Ik kocht een t-shirt van [AFC Ajax [Amsterdam]$^{\#LOC}$]$^{\#ORG}$.
   [Zondag met [Lubach]$^{\#PER}$]$^{\#OTH}$ is soms best vermakelijk.

6. Titles, salutations and owners do not belong to a complex proper noun. Owners can be proper names themselves as well.

   We luisteren naar [Lizzo's]$^{\#PER}$[Coconut oil]$^{\#OTH}$.
   $Meneer$[$Jansen$]$^{\#PER}$ vergat zijn hond in het park.

7. Proper names can occur as part of a complex token. Here, the entire token is annotated to contain a proper name.

   De verkoper stuurde mij een [whatsapp-berichtje]$^{\#WEBAPP}$.
   In een [facebookoproep]$^{\#WEBAPP}$ zag de fiets voorbij komen.
   Zonder [USB-C poort]$^{\#OTH}$ kon de mobiel niet worden verbonden.

8. If an NP cannot be contextually determined as a proper noun or appellative, it is not marked as NE.

   De stadspoort is een populaire ontmoetingsplaats.

9. If a name has become the name of certain items in the language and does not function as a NE in its use, it is not annotated.

   [Luna]$^{\#PER}$ hervond de teddybeer achter de bank.
   Voor haar verjaardag kreeg [Ritu]$^{\#PER}$ een bullet journal van haar moeder.

10. In the case of enumerations using hyphens or a postponement of a part of the NE to later words, the preceding NE-part is annotated as if it were written out in full.

    De [Eerste]$^{\#OTH}$ en [Tweede Kamer]$^{\#OTH}$ vergaderen tijdelijk digitaal.
    De provincies [Noord-]$^{\#LOC}$ en [Zuid-Holland]$^{\#LOC}$ werden opgesplitst in [1840]$^{\#DATE}$.

To which semantic class does a proper noun belong?

See semantic class tables with examples per class for clarification.
NE containing spelling mistakes, should still be annotated.

- Years in #ORG are not marked:

  [Janelle Monáe]$^{\#PER}$ trad op op [North Sea Jazz Festival]$^{\#ORG}$ [2019]$^{\#DATE}$.

- If a NE co-occurs with another word(part) as one token, label the entire token:

  [OCP/J. Slingers]$^{\#PER}$
  [marktplaats/emailaccount]$^{\#WEBAPP}$
  Hij heeft toegang tot mijn [marktplaatsaccount]$^{\#WEBAPP}$

- All parts of an address are labelled separately.
  [Herenstraat 12]$^{\#LOC}$ [1234AB]$^{\#LOC}$ te [Amsterdam]$^{\#LOC}$ [NEDERLAND$^{\#LOC}$

- Organisations that are a location within the context are label as such.
  Ik was in de [AH]$^{\#LOC}$

Appendix D

Semantic class tables

Label	Examples	
#PER	first name	Barack,
	intials	B.H.O., A.O.C.
	middle names	Hussein
	last names	Obama, Ocasio-Cortez
	full names	Barack Hussein Obama II
Sublabel		
#USER	username	bho2_196, aoc1980

Label	Examples	
#ORG	companies	Addias
	intitutions	Naturalis, KNMI
	acronyms	WHO, NOS
Sublabel		
#WEBAPP	website	Marktplaats, www.google.com
	social media	Instagram, facebook, Whatsapp, snapchat
#EDU	educational institution	Wuhan University, Basisschool Bohemen

[1]places of temporary nature such as festivals and venues are included
[2]days of the week e.g. *Maandag* are not annotated

Label	Examples	
#LOC	address	Stationweg 1, Den Haag
	zip Codes	2110 FJ
	place	Den Haag, Katwijk
	country	Zuid Holland, ZH
	local Region	Bollenstreek, Leiden Noord
	country	Duitsland
	places of entertainment, information and services[1]	
	entertainment venue	Duinrell, Snowworld, Haagse Kermis, Parkpop, Madurodam
	libraries and Museums	KB, Naturalis, Gemeente Museum
	park and Gardens	De Hoge Veluwe, Hortus
	other	De Haagse Markt, Blokker, IJspaleis

Label	Examples	
#MAIL	email address	example@domain.com
#PHONE	phone number	+31612345678

Label	Examples	
#DATE[2]	date of birth	12 Juli 1978, 12/07/1978,
	day	12de
	month(+)	Augustus, 12 Juni
	year	2008
	other dates	8 August 2011, 06-07-09
	holidays	Eerste Kerstdag, Dierendag

Appendix E

Evaluation matrix of labels

	precision	recall	f1-score	support
#BANKNR	0.92	0.83	0.87	53
#CODE	0.79	0.75	0.77	162
#DATE	0.97	0.96	0.96	283
#EDU	0.00	0.00	0.00	1
#KENT	0.84	0.80	0.82	20
#LOC	0.72	0.72	0.72	482
#LOCderiv	1.00	0.37	0.54	46
#MAIL	0.97	0.88	0.92	32
#ORG	0.52	0.48	0.50	159
#OTH	0.00	0.00	0.00	17
#PER	0.76	0.79	0.77	838
#PHONE	0.91	0.98	0.94	51
#USER	0.94	0.88	0.91	34
#WEBAPP	0.87	0.77	0.82	152
micro avg	0.78	0.77	0.77	2330
macro avg	0.73	0.66	0.68	2330
weighted avg	0.78	0.77	0.77	2330

Table E.1: Evaluation matrix for each individual label for the best context preserving model (the fourth row of Table 4.13).