Master Text Mining

Two Dutch fine-tuned BERT models:

Named Entity Recognition and Named Entity Linking to increase findability of local geographical information.

J. van Vugt (2598523)

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
Second Reader: Isa Maks

Abstract

Statistics Netherlands provides all statistical information of the Netherlands, however the search engine still needs improvement on findability of this statistical information. An agile team was set up to provide in such a solution and created a new search engine called CerBeruS. At the moment, incorrect information appears whenever a user is searching for specific local geographical information in the CerBeruS search engine. Therefore the goal of this internship project was to improve the findability of local geographical data in the CerBeruS search engine.

Several methods were applied during this internship project. To create annotations, the first step was to retrieve documents containing local geographical information from the OpenData source of Statistics Netherlands. This retrieval was done by creating a rule-based system that only selected documents containing local geographical information. The task that was performed for this internship project was a Named Entity Recognition and Named Entity Linking task.

A Named Entity Recognition task was performed by fine-tuning two Dutch BERT models, BERTje and RobBERT. In order to disambiguate ambiguous words, a Named Entity Linking task was performed. In this way local geographical mentions were linked with the corresponding local geographical code from the gazetteer.

The results show that both BERTje and RobBERT have a high performance score for Named Entity Recognition. Which entails that Named Entity Recognition performs well on its own. Disambiguating difficult cases is done by combining both Named Entity Recognition and Named Entity Linking, which shows to perform well.

Declaration of Authorship

I, Jasmine van Vugt, declare that this thesis, titled Two Dutch Fine-tuned BERT models 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: June 29th, 2021

Signed: 

Acknowledgements

During this internship project I have received a lot of support. First I would like to mention that I am very thankful for all my CerBeruS colleagues of Statistics Netherlands who made this online internship a good one. From the first day I felt welcomed and involved in the team. In specific I would like to thank Piet van Dosselaar of Statistics Netherlands, who gave me the opportunity to do this interesting and educational internship at Statistics Netherlands.

Furthermore, I wish to give my special thanks to my supervisor Sophie Arnoult of the Vrije Universiteit who really supported me during this internship. I want to thank you for critically checking my texts, which gave me the opportunity to improve this master thesis. Furthermore I wish to thank you for the guidance throughout each stage of the process.

I would also like to thank Lucas Lageweg of Statistics Netherlands for helping me with technical constraints I encountered and for supporting me to apply new techniques using BERT.

Another special thanks to my colleague Karen Goes, who supported me with the preparations of the internship presentation and checking my thesis for further improvements.

Thank you, without your support this internship project would have not been the same.

Jasmine van Vugt

Table of Contents

1. Introduction

1.1 Problem Setting and Approach

Nowadays a tremendous amount of digital documents are available. Searching on a website for a specific document seems straightforward, since most of the time a user finds what they were looking for. However, there is a lot that needs to be set for a user to find the correct digital documents. For instance, whenever a user wants to know what the criminality rates were in a specific year in a specific region it is important for the search engine to find hits that meet those criteria.

The goal of this internship project was to improve the search engine of Statistics Netherlands (CBS) for local geographical data. Statistics Netherlands provides all statistical information of the Netherlands on grounds of European Law. It is an organization that has a legal obligation to provide trustworthy statistical data and datasets that are freely available for everyone. All statistical information is accessible through the Open Data[1] of Statistics Netherlands and the datasets are available through StatLine. Due to the growing amount of data at Statistics Netherlands, as well as the demand for new techniques and methods to deal with this data, a new team is set up. This team created CerBeruS, a new search engine to increase findability of Statistical Information of the Netherlands.

The CerBeruS search engine still needs improvement when it comes to finding correct local geographical information. Whenever one searches for something that includes geographical information e.g. '*criminaliteit in Leiden'* (criminality in Leiden (city)) in the open data of Statistics Netherlands, multiple articles show up as a hit even though the articles are not about the city of Leiden. This is caused by the fact that in Dutch Leiden is a city name but also a verb. Therefore, articles that use the verb '*leiden*' (to lead) in the text are a hit when searching for criminality in Leiden. Right now the problem of the search engine is that precision of the hits are low whenever a user is looking for local geographical information.

1.2 Method

Previously, the approach of Statistics Netherlands consisted of vectorizing the documents and applying TF-IDF to obtain correct local geographical information. During vectorizing words are put inside a vector space were each word represents a number, based on the syntactic and semantic relationship in that vector space (Mikolov et al., 2013). TF-IDF is an approach used in text classification to rank documents based on relevancy with the input query (Paik et al., 2013). Also, an additional method was added: bag-of-words. When applying bag-of-words, a text is treated as a set of words (n-grams) (Li et al., 2016). Word order of sequences are removed because often it does not need to be taken into account. Also, bag-of-words is easy to use. Statistics Netherlands uses this bag-of-words method because of its

[1] https://opendata.cbs.nl/

effectiveness and its popularity for the text classification task. They applied TF-IDF, which works well with bag-of-words, to find a matching category for documents (Yun-tao et al., 2005). It was expected with vectorizing in combination with TF-IDF and bag-of-words, that with for example, 'Noord-Brabant', the system behind the search engine would pick this word up and rank relevant documents according to cosine similarity. But this approach did however not work as expected, since the documents often did not contain information that was searched for.

To handle the abovementioned problem, Text Mining approaches are a solution. Text Mining provides the opportunity to structure and order a big amount of digital documents. There are various approaches for structuring digital documents and increasing findability. Document and text classification is a well-studied approach to increase findability of documents where users search for and to categorize the collection of documents (Gayathri & Marimuthu, 2013). Named Entity Recognition (NER) and Entity Linking (EL) are well-known tasks in Text Mining. NER makes it possible to extract entities from texts (Goyal et al., 2018) and EL links those entities to a corresponding entry in a knowledge base (Yamada et al., 2015). NER cannot disambiguate while EL can, meaning that EL only links entity codes to actual entities. Therefore, a combination of applying NER and EL shows to perform well (Yamada et al., 2015; Martins et al., 2019). To extract local geographical entities I decided to use NER for this internship project, for disambiguation of entities I decided to apply EL. Therefore, the task of this internship project can be regarded as a NER/EL task.

Thus, for this internship project I linked local geographical indications within articles with their corresponding local geographical codes, by applying NER and EL. The local geographical codes were retrieved from the gazetteer, a document containing all of the place names of the Netherlands with their corresponding place name code [2].With NER local geographical indications are classified as being an entity or not. With EL those entity mentions are linked to a corresponding local geographical code from the gazetteer. We expected the precision of the output of the search engine to improve. So the search engine of CerBeruS will no longer give 'leiden' in the sense of 'to lead' as output whenever one searches for criminality in Leiden. Thus improving the findability of local geographical data.

As indicated in the previous paragraph, a NER and EL task was performed for this internship project. Through NER geographical entities are extracted from the articles. Different NER labels are used to resolve ambiguity. I decided to use state-of-the-art techniques for NER, by manually annotating local geographical entities and generating predictions whether an article of Statistics Netherlands contains local geographical information and what entity it is about. These NER predictions are generated for two Dutch pre-trained BERT models since earlier results on BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) shows that BERT models provide high results on NER. After obtaining the predictions, EL

[2] https://drive.google.com/file/d/11JN0vtd4vYSZmqhV29f7GprWYc6Fc9WD/view?usp=sharing

was performed to disambiguate difficult place names that could have multiple meanings and therefore have different local geographical codes. This was done by linking the local geographical code to a matching entity, based on the predictions.

1.3 Research Question

As stated before, Statistics Netherlands applied TF-IDF in combination with Bigrams to find local geographical data in the search engine. However, this approach did not work as expected and findability of local geographical information is still low. Therefore the research question of this internship project is:

Will findability of local geographical data in the CerBerus Search increase when applying Named Entity Recognition in combination with Named Entity Linking?

As set out earlier, NER prevents false matches to occur, because of the different entity labels that are used. Only entities that indicate actual local geographical information are labelled. Therefore 'leiden' in sense of 'to lead' will not be labelled, but 'Leiden' in sense of the municipality does receive a label. For this NER task fine-tuning is performed with two Dutch Language models, BERTje and RobBERT. Two Dutch language models are selected since the data used for this internship project is standard Dutch. Advantages of using pretrained language models for NLP tasks is the allowance of fast and reliable application of NER. Also, pretrained BERT models perform well on little data because of the sufficient amount of pre-training (Devlin et al., 2018). Furthermore, using pretrained language models for NER is the state-of-the-art. BERTje and RobBERT were used for the NER task because of the fact that both models are Dutch pretrained language models. Both pretrained models were fine-tuned on a similar fine-tuning regime to compare the two models and get insight into which model performs better. In this way Statistics Netherland is able to select the model that has highest performance on NER for standard Dutch data.

After fine-tuning the two BERT models, EL was applied. EL was used to decrease ambiguity of difficult cases even more, by linking the corresponding code based on the predictions. So, when 'Utrecht' was predicted as a municipality, the municipality code of 'Utrecht' was added to this entity. This leads to better findability of local geographical information a user is searching for. As discussed in the previous paragraph, during annotations different entity labels are used to make a distinction between the different entity types. All entities included local geographical indications, however different categories of those indications are used. By applying different labels, disambiguation was possible. Based on the predictions with different labels, EL could be applied in such a way that a specific entity, containing a specific label was linked to that specific local geographical code as in the example of the municipality 'Utrecht'. By doing this only articles with that local geographical code would show up if a user searches for 'municipality Utrecht'.

1.4 Outline

The thesis is structured as follows:

- Chapter 2 Theoretical Background:
    1. Text Classification and TF-IDF
    2. Named Entity Recognition with Pretrained Language Models
    3. Entity Linking
- Chapter 3 Method:
    1. Preparing NER
    2. BERT
    3. Entity Linking
- Chapter 4 Results:
    1. Results Fine-Tuning BERT
    2. Results on Named Entity Linking
- Chapter 5 Discussion:
    1. Method & Results
    2. Recommendations and Future Work
- Chapter 6 Conclusion

1. Theoretical Background

As set out before in the Introduction, CerBeruS uses TF-IDF in combination with bigrams and vectorizing. This approach does not work sufficiently because irrelevant documents show up when searching for specific information. In Section 2.1 the theoretical background of the CerBeruS models will be discussed. In Section 2.2 the theoretical background of NER tasks is set out, a task I decided to apply for this internship project to extract local geographical entities from texts. To generate predictions of NER, I used two pre-trained language model. More information on language models is presented in Section 2.3. Finally, to disambiguate entity mentions for this internship project, EL is applied to link local geographical code with the corresponding local geographical entity. Theoretical information of EL is discussed in Section 2.4.

2.1 Text Classification

As discussed in Chapter 1, the amount of digital textual information is growing and there is a need for methods to deal with this amount of data. Text Mining provides such a solution by extracting important information from structured, semi-structured and unstructured texts (Feldman & Dragan, 1995).

Text Classification is a Text Mining task in which natural language texts are labelled by predefined categories (Sebastiani, 2002). For this internship project Text Classification was necessary to identify local geographical categories in texts. This way a supervised machine learning algorithm is able to classify categories on new texts by training on its predefined training set. An advantage of this approach is that the amount of time needed for manually labelling texts decreases. Furthermore, Text Classification provides a solution for organizing texts in predefined classes and makes it easier to extract information from texts (Agarwal & Mittal, 2012).

TF-IDF is one of the methods applied for Text classification as a part of information retrieval systems. Documents are ranked based on relevance when encountering a given query (Paik et al., 2013). In TF-IDF, TF stands for the frequency of the word occurring in a document and IDF stands for the fact that the frequency is inversely proportional. Meaning that whenever a specific word occurs more often in a given text, the importance of other words that occur decrease. By using TF-IDF each word in a document is weighted on its uniqueness. In its turn by applying TF-IDF, relevancy between words and text documents is captured (Yun-tao et al., 2005). The formula for calculating TF-IDF is:

$$wd = fw, d * log (|D|/fw, D) (2),$$

In which ‘w’ indicates words and ‘d’ indicates documents. The ‘fw, d’ part stands for frequency of ‘w’ occuring in ‘d’. ‘|D|’ indicates the size of the corpus and ‘fw, D’ looks at the frequency of ‘w’ occuring

in ‘D’ (Salton & Buckley, 1988; Ramos, 2003). Common used words, such as pronouns and prepositions receive a low TF-IDF value based on this formula (Ramos, 2003).

In addition to TF-IDF, Statistics Netherlands used bigrams. Bigrams is a method used to capture words containing multiple subwords. While applying TF-IDF in combination with bigrams documents are ranked based on relevancy of the query and the documents. A method for ranking documents on relevancy is cosine similarity. An additional method that was used while applying TF-IDF is to apply vector space models which vectorise queries and documents (Paik et al., 2013). Using this method the vector of the query is compared to the vectors of the documents and their similarity is measured with a cosine function. This method combines three factors while computing the weight of the word. First, the density of a word inside a document, second the occurrence of a document containing the given term and third the span of these documents (Paik et al., 2013).

Advantages and Inconveniences TF-IDF

Table 1 shows the current performance of the CerBeruS Search Engine for finding local geographical information a user is looking for.

	Local Geographical information searched:	**Excerpt from Article**	**Actual use of Entity**
1)	Leiden (Municipality)	Dit paper beziet twee verschillende benaderingen om duurzame ontwikkelingsindicatoren af te ***leiden*** uit een systeem dat is gebaseerd op de Nationale Rekeningen. Deze complementaire methodes maken het mogelijk om de verschillende dimensies van duurzame ontwikkeling in onderlinge samenhang te analyseren en te evalueren. [3]	'leiden' in sense of 'to derive'
2)	Noord-Brabant (Province)	Deze tabel bevat cijfers over het onderwijsniveau en sector van werk van personen van 15 tot 75 jaar in *Noordoost-Noord-Brabant* (Corop-gebied).[4]	Noordoost-Noord-Brabant (COROP)
3)	Noord-Brabant (Province)	In deze maatwerktabellen heeft het Centraal Bureau voor de Statistiek (CBS) cijfers samengesteld over de internationale handel in goederen van de regio *Zuidoost-Noord-Brabant*. In deze maatwerktabellen staat informatie over de goederenhandel van de regio *Zuidoost-Noord-Brabant*. De goederenhandel is per regio in euro's gegeven. De cijfers hebben betrekking op verslagjaar 2018. Ook zijn de top 5 COROP gebieden in termen van export bepaald. De maatwerktabellen zijn geproduceerd in opdracht van de Brainport Development.[5]	Zuidoost-Noord-Brabant (COROP)
4)	COROP Utrecht	Er zijn wel sterke regionale verschillen in het niveau van de verkoopprijzen ten opzichte van 2008. In de vier grote steden (Amsterdam, Rotterdam, Den Haag en (1) *Utrecht*) liggen de prijzen duidelijk boven die van tien jaar geleden. In grote delen van Nederland echter niet. In de 'Randstadprovincies' Noord- en Zuid-Holland, (2) *Utrecht* en Flevoland zijn de prijzen hoger. In alle andere provincies zijn de prijzen niet op het niveau van 2008.[6]	(1) Municipality (2) Province

Table 1: Output of CerBeruS Search Engine when searching for Local Geographical Information.

[3] https://www.cbs.nl/nl-nl/achtergrond/2004/24/accounting-for-sustainable-development
[4] https://www.cbs.nl/nl-nl/maatwerk/2018/21/werkzame-bevolking-noordoost-noord-brabant-onderwijs
[5] https://www.cbs.nl/nl-nl/maatwerk/2019/45/goederenhandel-zuidoost-noord-brabant
[6] https://www.cbs.nl/nl-nl/nieuws/2018/36/huizenprijzen-op-niveau-van-voor-de-kredietcrisis

As shown in Table 1, ‘Leiden’ (1) in sense of ‘to derive’ shows up in the current search engine when searching for ‘gemeente Leiden’ (municipality Leiden). It is expected with the method used by Statistics Netherlands that whenever a user searched for ‘Noord-Brabant’ (2)(3) in the search engine, articles containing the specific mention ‘Noord-Brabant’ would show up. However as shown in Table 1, articles containing additional words such as ‘Zuidoost-Noord-Brabant’ show up. Furthermore, when searching for ‘COROP Utrecht’(4), the search engine is not able to make a distinction between the different entity types. As shown in Table 1, information of the municipality ‘Utrecht’ and the province ‘Utrecht’ are stated in the article, but no information on the COROP ‘Utrecht’. Overall Table 1 shows that the performance of findability of local geographical information is currently not accurate.

From the results in Table 1, one can see Text Classification for the CerBeruS Search Engine still needs improvements. The method of using TF-IDF, bigrams and vectorization does not perform well. Therefore, for this internship project the selected tasks to label natural language texts for Text Classification were NER and EL. NER and EL are two fundamental Natural Language Processing (NLP) tasks for labelling entity categories. NER detects mentions of named entities in texts and EL applies a knowledge base ID to the texts (Martins et al., 2019).

2.2 Named Entity Recognition

2.2.1 NER Task

One of the focuses of this internship project was to extract local geographical entities from articles of Statistics Netherlands. Methods and techniques to extract meaningful information are fast growing. One method of doing so is to extract named entities from texts. Named Entities are mentions like names of persons, organizations and locations inside a text (Tjong Kim Sang & De Meulder, 2003). NER is a task to identify such named entities in texts and is often used for information extraction systems. The first NER task was during the Sixth Message Understanding conference in 1996, which was organized by Grisham and Sundheim (1996). It is still a popular task that has led to a lot of research. At first only English was well represented for NER. However, nowadays systems are developed for different languages as well. During the shared task of CoNLL-2002 Spanish and Dutch were dealt with (Tjong Kim Sang, 2002). For the CoNLL shared task-2003 an English and German NER system was created by different researchers using different techniques (Tjong Kim Sang et al., 2003).

NER plays a major role as a subtask in natural language applications such as Information extraction and information retrieval (Goyal et al., 2018). Therefore NER was suitable for this project. To create a system of high quality it is convenient to first apply a low level task such as NER (Martins et al., 2019). In the case of this internship project NER was useful since it enabled the fine-tuning of two Dutch

language models on manually annotated data. Furthermore NER was applied to generate predictions of named entities in the articles of Statistics Netherlands.

2.2.2 Challenges

NER is a task applied to varying fields and domains of research and can therefore help in extracting all kinds of entity information. Often there is not enough labelled data available to train a NER system and therefore manual annotations need to be performed which is time consuming (Tjong Kim Sang et al., 2003). Since NER is also domain-related, generating domain-related labels is even more complicated.

2.3 Former Methods

NER is a well-studied NLP task for which different methods were applied in the past to gain the highest performance. There has been a shift in NER systems, from applying handcrafted rules, lexicons, orthographic features and ontologies to systems that combined feature-engineering and machine learning (Yadav et al., 2019). Later on neural NER systems were created with a minimal amount of features (Yadav et al., 2019). Also, as stated in the previous Section there are challenges in creating manual created gazetteers. Therefore, there has been a shift from supervised NER to semi-supervised NER to unsupervised NER. This section provides a background of the methods used for feature-based methods and neural approaches.

2.3.1 Methods

Feature-based methods

Feature-based machine learning systems are created through a combination of features and machine learning systems. The systems are often trained on supervised data and therefore able to make predictions on new example input (Yadav et al., 2019). Machine learning systems that are most common for NER are Hidden Markov Models (HMM), Support Vector Machines (SVM), Conditional Random Fields (CRF) and decision trees (Yadav et al., 2019).

One of the features that was shown to perform well for NER is capitalization. Malouf (2002) used capitalization to see whether a word appeared first in a sentence or whether it appeared before a well-known last name. To see if a word appeared before a well-known last name, 13281 first names were collected at first. Other features that were selected which gained high performance were again capitalization, trigger words, previous tag prediction, bag-of-words and gazetteers (Carreras et al., 2002).

As stated in the previous Chapter, during the CoNLL shared task-2003 an English and German NER system was created in which sixteen teams participated (Tjong Kim Sang et al., 2003). The data that was used for this shared task consisted out of eight files including English and German. Training data,

development data, test data and unannotated data was provided for this shared task. In order to tune parameters, the development data and test data were separated so the systems were not optimized on the test data. Tjong Kim Sang et al. (2003) mentioned that the choice of the learning techniques and features are equally important. Fifteen of the sixteen teams used lexical features. Furthermore, the majority of systems also used part-of-speech as a feature. However, Tjong Kim Sang et al. (2003) concluded from all sixteen systems that no feature is specifically suitable for NER which is different from the results of Malouf (2002) and Carreras et al. (2002). From the CoNLL shared task-2003 one system by Florian et al. (2003) had the highest performance. This system used a Classifier combination of Maximum Entropy Models, transformations-based learning, Hidden Markov Models and robust risk minimization (Tjong Kim Sang et al., 2003).

Agerri and Rigau (2016) created a semi-supervised NER system with features. The features used were n-grams, lexicons, prefixes, suffixes, bigrams, trigrams and unsupervised cluster features from the Brown corpus, Clark corpus and k-means means clustering with word embeddings (Mikolov et al., 2013). Results showed that this semi-supervised NER system performed almost as high as the supervised systems (Yadav et al., 2019). Liu et al. (2015) also created a semi-supervised NER system with word embeddings, those results showed that this semi-supervised approach performed equally as high as state-of-the-art results for supervised systems.

Neural Network Approaches

Besides feature-based methods there was also a rise in neural network approaches. As stated before, semi-supervised embeddings were used in the feature based method. For the neural network approach, those semi-supervised word embeddings were however replaced with word embeddings created from unlabelled data through a unsupervised method with for example skip-grams (Mikolov et al., 2013). Those pre-trained word embeddings were shown to be important for neural network based NER systems (Habibi et al., 2017; Yadav et al., 2019).

Nowadays, neural network approaches for NER focus on the representation of a word inside a sentence (Yadav et al., 2019). There are different categories on which neural networks can be applied, namely on word level, character level, subword level or a combination of them. Using a neural network means that each input is represented as a word embedding in the Recurrent Neural Network (RNN).

While using a neural network on word level, words of a sentence are represented as word embeddings. Results of Huang et al. (2015) on this approach showed that adding a CRF layer on top of a Long short-term Memory (LSTM) model improved performance of the model. When using a neural network on character level a sentence is seen to a sequence of characters (Yadav et al., 2019). Each character is looked at by the RNN and labels are predicted for each character. Kuru et al. (2016) used character level architecture for NER on 7 different languages using a Viterbi decoder. Gillick et al. (2017) applied a

different approach using an encoder-decoder architecture in which each character was encoded in bytes. Overall results on the character architecture looked promising in comparison to the state-of-the-art performance.

Combining character architectures with word level architectures is shown to create a high performing NER system that does not need domain specific knowledge or resources (Yadav et al., 2019). The highest performing NER system for this combined architecture was created by Lample et al. (2016). In this system LSTM word embeddings are merged together above the characters of a word, then an additional sentence level is added to Bi-LSTM and an additional CRF layer is added for the prediction labels.

To conclude, neural network approaches result in higher performing NER systems than feature-based methods. A combination of word-based and character-based models performed better than models based solely on words or characters (Yadav et al., 2019).

2.3.2 Challenges

For the feature-based methods there are challenges. For example, research on NER by Ritter et al. (2011) focused on classifying named entities in tweets by using a feature-based method. They found that classifying tweets on named entities was a difficult task, caused by the great range of named entity types. Also, tweets have a limited length of 140 characters, therefore the context to determine entities was missing (Ritter et al., 2011). Results of NER on tweets showed that news-trained Named Entity Recognizers relied on capitalization, which is not a sufficient indicator in tweets (Ritter et al., 2011). To overcome this problem, 2400 tweets with named entities were manually annotated. Those annotations contained 10 of the most frequently used entity types on Twitter. Results showed that a model that was trained on the annotated training data of tweets, outperformed state-of-the art news-trained training data.

Another example of a challenge for feature-based methods was encountered in the study of Kim and Cassidy (2015). Kim and Cassidy (2015) performed NER to automatically identify person or location names from Historical Newspapers. The method applied was aimed at comparing a pre-trained Stanford NER with a Stanford NER system based on semi-supervised annotated training data. The test data consisted of manually annotated data, also known as the gold standard. An evaluation of the two NER systems on the test data showed that the Stanford NER with semi-supervised annotated training data did not perform better than the pre-trained Stanford NER system. Therefore, the pre-trained Stanford NER was applied, since annotating the amount of unannotated data of the semi-supervised Stanford NER was too time consuming.

Kim and Cassidy (2015) found that the mentioning of names inside texts showed some peaks in different time-periods. To determine and get insight into whether a mention of a name indicated the actual person

that was mentioned or a different person, clustering was applied to decrease ambiguity. NER does not disambiguate, meaning that it does not make a distinction between entities with the same mention. For example, for this internship project 'Utrecht' could mean a province, a municipality or a COROP but NER does not make a distinction between the three meanings of this word. To overcome this problem, Kim and Cassidy (2015) applied clusters by using vector-space word representations for disambiguation and used word2vec to generate these word vectors (Mikolov et al., 2013). Applying vector-space word representations is shown to improve NLP tasks (Collobert and Weston, 2008; Socher et al., 2013; Nguyen et al., 2015; Kim et al., 2015). As expected, results showed that those clusters give a clearer indication whether it was about a specific person or another person (Kim et al., 2015).

To conclude, there are challenges for capitalization and context while using feature-based methods. Also, it was shown by Kim and Cassidy (2015) that disambiguation for NER systems with only features is impossible. Therefore, word vectors were applied to disambiguate ambiguous words. This indicates that applying feature-inferring neural network systems show to be important while creating a NER system (Yadav et al., 2019).

2.4 Pre-trained Language Models

As indicated in the previous Chapter, neural network approaches are shown to have higher performance than feature-based methods and neural networks take context into account. Nowadays, neural networks are the state-of-the-art for NER. Those reasons led to selecting two pre-trained language models for this internship project.

For feature-based methods, word embeddings were originally used as word representations as discussed in Section 2.2.1 (Mikolov et al., 2013). However, those embeddings only captured single context representations for each word. Therefore, Language Models were used with vectors retrieved from a bidirectional LSTM. The first feature-based bidirectional Language Model that was created was ELMo, which stands for Embeddings from Language Models (Peters et al., 2018). Unlike word embeddings, ELMo was the first model that took context into account and added pre-trained representations as features (Peters et al., 2018; Devlin et al., 2018). However, ELMO uses a task specific architecture, while a fine-tuning approach uses minimal task-specific parameters. From this objective Devlin et al. (2018) decided to create a model named BERT.

2.4.1 BERT

BERT stands for Bidirectional Encoder Representations from Transformers. This model improves on earlier models by using a multi-layer bidirectional Transformer encoder (Vaswani et al., 2017). BERT Transformer makes use of bidirectional self-attention to take both left-to-right as well as right-to-left context as illustrated in Figure 1. Unlike BERT, ELMo makes use of independently trained left-to-right

and right-to-left LSTSMs for retrieving features for downstream tasks and OpenAI GPT uses only a left-to-right Transformer. Thus, for both OpenAI GPT and ELMo not all surrounding words are taken into account, while BERT takes all surrounding words into account and therefore has more knowledge of the context in which a word occurs (Devlin et al., 2018).

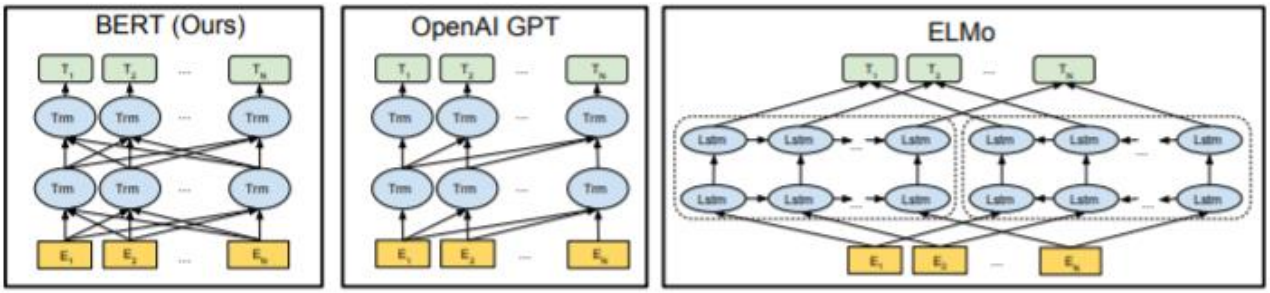

Figure 1: Retrieved from Devlin et al. (2018)

The framework Devlin et al. (2018) introduced consists of pre-training and fine-tuning. Pre-training is done on unsupervised data to create a language specific or multilingual language model that could later be fine-tuned on supervised data for a specific NLP task in a specific language (Devlin et al, 2018). After fine-tuning an additional output layer is added on top of the pre-trained model.

As explained before, Devlin et al. (2018) decided that during pre-training a BERT model takes the entire context into account. To do so, pre-training consists of two unsupervised tasks objectives namely Masked Language Modelling (MLM) and Next Sentence Predictions (NSP). Devlin et al. (2018) introduced MLM for pre-training to train a deep bidirectional Transformer. MLM randomly masks tokens from the input, the models predict the token that is masked by taking all surrounding words into account. The second objective is NSP and trains a model to understand sentence relations (Devlin et al, 2018). Data used for pre-training BERT consisted of the BooksCorpus (Zhu et al., 2015) and English Wikipedia.

Devlin et al. (2018) tested fine-tuning on 11 NLP tasks where the output of this fine-tuning approach added an additional layer to the pre-trained model. Results of this study showed that BERT outperforms all systems on all 11 NLP tasks. Furthermore, results of this study showed that extreme model sizes, which indicates that a lot of pre-training was done, works well on small scale tasks where the amount of data is little (Devlin et al., 2018). Indicating that a sufficient amount of pre-training improves the performance on a task when a limited amount of data is available.

2.4.2 Alternative Model: RoBERTa

After the development of BERT, Liu et al. (2019) wanted to adjust BERT with further improvements. Therefore a new model was created, RoBERTa, which stands for Robustly optimized BERT approach (Liu et al., 2019). This model was pre-trained with different objectives than BERT. As outlined in the

previous Section, BERT made use of NSP while Liu et al. (2019) of RoBERTa left NSP out. Furthermore, the model was trained longer on greater batches, more data, longer sequences and a different approach was applied for the masking objective (Liu et al., 2019).

RoBERTa was thus pre-trained differently from BERT. Instead of randomly injecting short sequences as with BERT, RoBERTa applied full-length sequences. Also, results of another study revealed that increasing the amount of data while pre-training, results in a better performance (Baeviski et al., 2019). Therefore, Liu and colleagues added additional training data to the RoBERTa model. As previously set out, BERT was pre-trained on two corpora, namely the BookCorpus and English Wikipedia. RoBERTa was pre-trained on five corpora, namely the BookCorpus (Zhu et al., 2015), English Wikipedia, CC-News, OpenWebText (Gokasland and Cohen., 2019) and the dataset Stories (Trinh et al., 2018). So in total RobBERTa was pre-trained on three corpora extra then BERT.

As mentioned above, the masking objective which was applied for the BERT model was approached differently for the RoBERTa model. BERT applied a static mask, indicating that random tokens were masked and predicted. RoBERTa used dynamic masking to ensure that each sequence was masked differently on every epoch while training. Just as for BERT, subwords were taken into account for RoBERTa with Byte-Pair Encodings (BPE) (Sennrich et al., 2016). BPE allows handling large vocabularies on both character as well as word level and extracts statistical analysis of subword units of the training corpus. Devlin et al. (2018) decided to use a character level BPE of 30K subword units for BERT while Liu et al. (2019) decided to use a BPE vocabulary which contained 50K subword units for RoBERTa.

Results of the dynamic masking showed slight improvements, which was therefore applied by Liu et al. (2019) for RoBERTa. In addition the NSP was removed from pre-training as well, since findings demonstrated that removal of NSP increased performance on downstream tasks (Liu et al., 2019). Furthermore, the batch size was increased for RoBERTa, which resulted in a better understanding of the masked language objective and improved the accuracy of a specific task (Liu et al., 2019). Altogether, results of the study by Liu and colleagues (2019) showed that a longer training time, adding more batches, additional training data and removing NSP resulted in improvement of the performance of the model (Liu et al., 2019).

2.4.3 Multilingual and Dutch Language Models

BERT was pre-trained on English corpora by Devlin et al. (2018) but the data I work with for this internship project is in standard Dutch. For other languages than English, one can use multilingual BERT. Multilingual BERT was pre-trained in 104 different languages including Dutch [7](Devlin et al.,

[7] https://github.com/google-research/bert/blob/master/multilingual.md

2018). However, a number of other of languages do have dedicated models, namely Italian (Polignano et al., 2019), French (Le et al., 2019) and Finnish (Virtanen et al., 2019; De Vries et al., 2019). De Vries et al. (2019) expected that a Dutch model performs better on a Dutch NLP task than a multilingual model and therefore presented the first BERT-based pre-trained language model for Dutch.

BERTje is a Dutch pre-trained model created by de Vries et al. (2019) and was pre-trained on high quality Dutch corpora (de Vries et al, 2019). Those corpora were, books, A Multifaceted News Corpus (TwNC), a multi-genre reference corpus (SoNar), web news from January 1, 2015 until October 1, 2019 and Dutch Wikipedia (de Vries et al, 2019). BERTje used a similar pre-training regime as for BERT with MLM and NSP. During the pre-training de Vries et al. (2019) however found that the NSP task did not work accordingly, therefore BERTje is trained on a Sentence Order Prediction (SOP) objective. The SOP objective is used to predict whether a sentence is either the next or previous sequence (de Vries et al., 2019). De Vries et al. (2019) decided to apply a different MLM than BERT, since they found that splitting words in the same way as BERT was too easy to predict for the model (Lan et al., 2019). With the adjusted MLM, not a single word piece was masked but sequential word pieces that belong to the same word were masked (de Vries et al., 2019).

De Vries and colleagues compared BERTje with multilingual BERT on NER, Part-of-Speech tagging, Semantic Role Labelling and Sentiment Analysis. Findings indicated that a Dutch BERT model improves state-of-the-art performance of NER. Furthermore, the results showed that for Part-of-Speech, Semantic Role Labelling and Sentiment Analysis, BERTje outperforms multilingual BERT.

As BERTje is a pre-trained Dutch language model for BERT, RobBERT is a Dutch pre-trained language model of RobBERTa. RoBERTa is pre-trained on English data. Models that are pre-trained on a specific language are shown to perform better on language specific NLP tasks than a multilingual model (Martin et al., 2019, de Vries et al., 2019, Delobelle et al., 2020). To perform Dutch NLP tasks, RobBERT was created by Delobelle et al. (2020) containing a Dutch dataset to increase performance.

Delobelle et al. (2020) applied similar training regimes as for RoBERTa during pre-training of RobBERT. For RobBERT two versions were created, one that only changed the datasets (RobBERT v1) and the other one replaced the datasets and the tokenizer (RobBERT v2). The data used for pre-training was the Dutch part of the OSCAR Corpus and consisted of 39GB of Dutch data. This amount of Dutch data is more than the data used for BERTje which contained only 12GB. For RobBERT v2 the BPE tokenizer was not applied, but Delobelle et al. (2020) created a Dutch tokenizer from the OSCAR corpus. Furthermore, similar training objectives are taken into account as used for the RoBERTa model, namely that only MLM is used in which a masked token in a specific location in a sequence is predicted.

Delobelle et al. (2020) evaluated performance of RobBERT on two Dutch NLP tasks: sentiment analysis and pronoun predictions. For the sentiment analysis they selected the Dutch Europarl corpus (Koehn.,

2005). Results on fine-tuning RobBERT on sentiment analysis showed that RobBERT v2 outperforms other language models on Dutch sentiment analysis (Delobelle et al., 2020). Furthermore, RobBERT was fine-tuned on the Die/Dat Disambiguation task (Allein et al., 2020 as cited in Delobelle et al., 2020). *Die* and *dat* are Dutch words that could exist as both demonstrative or relative pronouns. These could on top of that be used to introduce a clause in a sentence and is dependent on the gender of the referring word. To approach this task, words should be predicted in which the MLM is used. Furthermore, a masked sentence objective should be performed in which the model needs to decide from two sentences where *die* and *dat* are used accordingly. Overall Delobelle et al. (2020) showed that RobBERT v2 outperforms all other language models, therefore I decided to use the RobBERT v2 model for this internship project.

The study of Delobelle et al. (2020) revealed that RobBERT outperforms other language models on this task. The findings also showed that less data is needed for the model to perform better than other language models. On top of that RobBERT has more knowledge on Dutch than any other model (Delobelle et al., 2020). This might be caused by the different pre-training and the larger Dutch corpus used in training the model. Both BERTje and RobBERT are used for this project, since the data used in this project is – like the data used for the two pre-trained language models - is standard Dutch. In this way a comparison between the performance of BERTje on the one hand and RobBERT on the other hand is made, since as indicated before, the pre-training was different for both models.

2.4.4 Remaining Challenges

There are still remaining challenges for BERT models. For example Akbik et al. (2019) found that rare words are often underspecified. This problem was approached by making the model memorize the contextual representation of a unique word or string and create contextual embedding which could be used for classification. Another approach was to use each first sentence to create a prediction (Virtanen et al., 2019).

Furthermore, BERT does not solve problems in limited amounts of availability of task specific training data. As stated earlier, manual annotations are time consuming and therefore different methods are created to automatically generate labelled data (Liang et al., 2020). But Liang et al. (2020) discovered that during automatic creations of labelled data, annotations were missing.

Also, during automatic retrieval of labelled data an induction method of popularity is often applied (Liang et al., 2020). As for this internship project ‘Utrecht’ could be a municipality, COROP or a province and could therefore be mapped to multiple entity types. With the induction method, a bias for matching was created for popular types in the data (Liang et al., 2020). So whenever municipality ‘Utrecht’ is most popular it might occur that COROP ‘Utrecht’ and province ‘Utrecht’ will be labelled

as municipality. Because of this wrong annotations, it seems like performance of BERT is high while in fact it is not able to make distinct predictions and a lot of false – positives occur (Liang et al., 2020).

2.5 Named Entity Linking

NER is an NLP task used to extract entities from texts and label these entities with predefined labels. As indicated in the previous chapter, there are still challenges for fine-tuning BERT on NER because of the ambiguity of words. Mentions such as 'Utrecht' can have multiple meanings, however NER does not take those differences into account. Table 2 shows examples of polysemous words with the different local geographical codes used by Statistics Netherlands and should therefore be labelled differently.

Local Geographical Entity	**Polysemy**	**Polysemy**	**Polysemy**
Leiden	Municipality (GM0546)	'To lead' No code	
Huizen	Municipality (GM0406)	Plural of house No code	
Heel	Municipality (GM1937)	'Really' No code	
Utrecht	Municipality (GM0344)	COROP (CR17)	Province (PV26)
Groningen	Municipality (GM0014)	Province (PV20)	

Table 2: Polysemous Words that cause Ambiguity

To solve ambiguity, EL is often applied. Entity Linking (EL) is an NLP task performed to detect entity mentions in a text and link those entity mentions with a corresponding entry in a knowledge base (Yamada et al., 2015). Different methods of applying EL are used, namely two-step approaches and combined approaches. These approaches will be explained in more detail in Section 2.5.1 and Section 2.5.2.

2.5.1 Two-Step Approaches

In a two-step approach NER and EL are performed separately. A two-step approach on EL was performed by Yamada et al. (2015). The main objective of this research was to increase the performance of NER on tweets. Tweets are shown to be noisy, which causes NER software to perform worse. Yamada et al. (2015) performed this EL task by recognizing a set of entity mentions and matching those mentions with a corresponding referent entity inside the knowledge base. Those entities were extracted by similarity of mentions that were greater than a 0.9 soft TF-IDF threshold. This way Yamada and colleagues created a dictionary of pairs of mentions and referent entities.

At first a NER task was performed to detect entity mentions or non-entity mentions. Yamada et al. (2015) approached this by labelling n-grams, instead of using regularly used BIO labelling and applying supervised machine learning by combining the NER output and the EL with entity mentions. Results of this research showed that knowledge bases improve NER by applying EL. Furthermore, results of this research showed that this method is effective for both a segmentation as well as a classification task (Yamada et al., 2015).

2.5.2 Combined Approaches

In comparison to the two-step approach, the combined approached performs NER and EL simultaneously. Research on NER and EL performed by Martins et al. (2019) involved this combined approach of NER and EL. Martins et al. (2019) trained and evaluated NER and EL on the AIDA/CoNLL dataset (Hoffart et al., 2011) and labelled entity mentions as person, location, organization and miscellaneous. Whenever a mention is identified by NER, EL is performed through disambiguating these mentions by linking this to the knowledge base. This study applied a bidirectional LSTM to take context into account for NER (Martins et al., 2019). Martins et al. (2019) compared this combined approach with the state-of-the art NER models and EL models. This comparison showed that this combined approach increases the performance of both NER and EL (Martins et al., 2019).

Altogether, from previous research one can conclude that applying EL for NER improves the performance of a system and makes disambiguation possible. The data used for this project is ambiguous as well, therefore EL was found suitable for this classification task. It is expected when applying EL for this project that whenever a text mentions ‘Leiden’ as a municipality, it is indeed labelled and linked as a municipality. An example of this can be seen below:

(1) Wij *leiden* een druk leven.

(2) De stad *Leiden* is een echte studentenstad.

In Example (1) ‘leiden’ is used as a verb, whereas in the second example (2), the municipality ‘Leiden’ is mentioned. Only the second mention is linked with a corresponding local geographical code and will

show up if one searches for information on the municipality Leiden. Using this method ambiguous words are disambiguated.

2.5 State-of-the-art NLP Tools

As discussed in the previous Section, BERTje and RobBERT are two pre-trained models which could be applied to a range of tasks. For this project those models were retrieved from Huggingface, an open-source library which supports the use of pre-trained models and makes use of the Transformers architecture (Wolf et al., 2020). Transformers is the current state-of-the-art architecture for NLP. With this architecture one is able to pre-train on large text corpora which increases accuracy on downstream tasks such as text classification (Yang et al., 2019).

The design of this architecture is what is normally used for NLP machine learning pipelines as well (Wolf et al., 2020). As a first step data is processed, then a model is applied and from this model predictions are generated. As explained earlier, tokenizers are applied for the BERT models which is a crucial NLP-specific aspect. The tokenizer stores classes into a token-to-index map and performs encoding and decoding according to the process of tokenization initiated by a specific model. Then sparse indices are transformed to contextual embeddings. After that a head is created from the textual embeddings of which task-specific predictions are made. The heads are used during fine-tuning and put on top of the contextual embeddings of Transformers as output layer.

Overall, Huggingface is indicated as being an important library for state-of-the-art NLP. By using the Transformers architecture and the pre-trained models, task specific models can be fine-tuned through which an additional layer is added on top of the pre-trained model. To fine-tune BERTje and RobBERT for NER I therefore decided to use the Huggingface Library.

3.Method

As outlined in the previous Chapter, BERT performs well for NER and NER in combination with EL make disambiguation possible. This chapter describes the method for preparing NER, BERT and EL for this internship project. Section 3.1 provides information on preparing NER. Section 3.2 provides the method used for applying BERT. Finally, Section 3.3 presents the method of EL for this internship project.

The task of this internship project was to perform NER and EL based on local geographical entities that are mentioned in the documents. First NER was performed to extract local geographical entities from the text. After that those entities were linked with the related local geographical codes. To perform NER a rule-based system was created to select files containing local geographical information. Afterwards manual annotations were performed. Finally BERTje and RobBERT were used to make predictions on Named Entities.

3.1 Preparing NER

This section presents the data and the method used while preparing NER. The method used to create a rule-based system through which documents including local geographical information were retrieved is explained. After that manual annotations were performed to create a gold standard for NER.

3.1.1 Data

The data used for this internship consisted of 2771 articles of the open data of the CBS. The total amount of words was 250742 and the average amount of words inside a document was 90 words. An example of an article can be found in Appendix I. This article contained 195 tokens. To find local geographical information inside those articles, a gazetteer was used. This gazetteer contains names of rural areas, provinces, municipalities and ‘COROPs’ (areas based on statistical grounds) and their corresponding regional codes.

3.1.2 Selecting Data for Annotations

This section presents the method for creating the rule-based system. A rule-based system was created to get familiar with the data and get an understanding on which pre-processing steps to carry out to normalize for string matching. Furthermore, the rule-based system retrieved documents that contained local geographical information. By doing this the amount of documents for annotations decreased, since not all documents contained local geographical information.

Pre-processing for Normalization

The first step of this internship project consisted of pre-processing to be able to normalize names and apply string matching for the articles and the gazetteer. Within the gazetteer there was additional information included for some local geographical indication. For rural areas it was (LD), for provinces (PV), for some municipality names (gemeente, drostambt, oud) and 'COROPs' (CR). This additional information was removed, since in the articles those names are never mentioned with this additional information. However, there were some difficult cases in which the place name could be both a province and a municipality. Examples of such names are 'Groningen' and 'Utrecht'. Therefore, it was important to take context into account to acquire knowledge whether it was mentioned as a province or as a municipality.

A further pre-processing step was to remove '-' from, for instance, 'Noord-Holland'. The reason for removing this was a lot of variance with the articles on how such words could be spelled. To make sure both the local geographical data as well as the article data had the same spelling it was removed from the gazetteer as well. Furthermore, words written in Frisian, a second language which is spoken in the province Friesland in the Netherlands, has its own spelling and had to be added to the gazetteer manually. For example, in Dutch the province is spelled like 'Friesland', but in Frisian it is spelled as 'Fryslân'.

Difficult Cases in Normalization

For the articles the sentences were tokenized in which the text was segmented into sentences. The reason for choosing this method is because there are multiple indications of regions that consist of multiple tokens. An example of this is 'Agglomeratie Leiden en Bollenstreek'.In case word tokenization would be applied the machine would think that there are four tokens 'Agglomeratie', 'Leiden', 'en', and 'Bollenstreek'. Since 'Leiden' is also a municipality name it would be identified as a municipality, while in fact it is a 'COROP' name. For the pre-processing it was decided to leave the casing in its original state. One of the main reasons behind this was that there are local geographical indications that could also be adverbs, nouns or adjectives. For instance the adverb 'heel', could also be the place name 'Heel', or the noun 'Huizen' could mean houses or the place name 'Huizen'. By applying this method words that are not local geographical indications would most often already be all in lowercase, with the exception of the start of a sentence. Local geographical indications would on the other hand also already be uppercased since place names are always indicated with capitals.

Regular Expressions

To take the previous and preceding token into account regular expressions were created. The expressions that were used are the following:

(3) re.sub, r'(\/|\.)', r'\\\1'

(4) fr"\b({'|'.join(sorted_placenames)})\b")

For normalization the expression in (3) was used, which indicates that words containing a '/' or a '.' are replaced with '\/' and '\.'. The reason for doing this is because both '/' and '.' have a syntactic meaning in regex. By replacing this with r'\\\1 the backslash and dot are interpreted as a literal characters. The expression in (4) takes all sorted place names, puts it together and separates the place names with OR. The "\b and \b" indicate that only entire words are taken into account, only words within word boundaries are found. Other words, which are a part of a word are not taken into account while using this regular expression.

So, Regular expressions are used to specify a string and look at a pattern whether a string matches with another string or whether it does not match (Kaur, 2014). The local geographical indications are sorted based on length to make sure the regular expression takes 'Agglomeratie Leiden en Bollenstreek' into account instead of only 'Leiden'. If this entire string occurs in the article it is matched as a REGION for the rule-based system.

Rule-Based System output

Finally, the data returned by the rule-based system consisted of the sentence in which the place name occurred, the token, the place name itself, part-of-speech, and a label (REGION) that indicated that a place name was mentioned. An example of what this looks like can be found in Appendix II. It was decided to add part-of-speech because, as mentioned earlier, some local geographical indications in Dutch could also be for example adverbs, adjectives or nouns that do not necessarily mention a place name. This way the ambiguity was expected to become smaller. So, overall with the rule-based system only documents were selected that contained the region label. By doing this manual annotations included documents containing local geographical information or ambiguous words such as Heel.

3.1.3 Annotations

The rule-based system retrieved documents that contained local geographical information. To ensure that the documents only included local geographical information, manual annotations were performed using Inception. Inception is an environment that allows users to create annotations (Klie et al., 2018). Within Inception users can create a project which consists of documents to be annotated. Users can also create customized labels for the annotations.

For this internship project four different customized labels were created, namely LD (rural areas), PV (Province), CR (COROP), and GM (municipality). Inception provides the BIO schema automatically in which the beginning, inside or outside of a token are indicated. For instance, the COROP 'Het Gooi en Vechtstreek' annotation would be 'Het' (B-CR), 'Gooi' (I-CR), 'en' (I-CR), 'Vechtstreek' (I-CR). Tokens that belong together are labelled as one local geographical indication.

The gazetteer was used as a guideline to annotate the correct labels to the correct geographical indication. As stated before, there are difficult cases in Dutch such as 'Groningen', which could be a province or municipality. From context and surrounding words I could capture the correct label. Whenever a geographical indication indicated a municipality or province this was clearly stated in the text.

During annotations it was found that a lot of surnames, which also could be local geographical information, were selected by the rule-based system as being local geographical information. However this should not be labelled as local geographical information, since it does not state a rural area, province, COROP or municipality. Examples of such surnames were: 'van der Hoeven', 'de Vries', 'Bunschoten', 'van den Brakel', 'Kralingen', 'Rietveld', 'van Beuningen'. The same goes for different University names such as 'Universiteit Maastricht', 'Universiteit Utrecht'. Also, Airport names such as 'Airport Eindhoven', 'Luchthaven Groningen'. All those cases were not labelled as containing local geographical information, since it indicates names or organizations instead of actual local geographical information. Further annotation guidelines can be found in Appendix III.

In total 1024 documents were manually annotated. The total amount of tokens was 86351 and the average amount of tokens per document was 85. The total number of annotated labels are listed below in Table 3. As one can see for the entity labels, 'PV' received the most labels during annotations and 'LD' the least.

Labels	**GM**	**PV**	**CR**	**LD**	**Total Amount of Entities**	**Total Amount of Tokens**
Total	3657	4223	897	140	8917	86351

Table 3. Total Number of Annotated Labels

3.2 BERTje and RobBERT for NER

As mentioned earlier, two BERT models were applied for NER, since fine-tuning BERT is the current state-of-the-art for NER (Devlin et al., 2018). This section provides the method applied for those two models and the experimental set-up.

3.2.1 Models

To create predictions for NER, two BERT models were used. The two Dutch based BERT models selected were BERTje and RobBERT. It was shown that single language models perform better than multilingual models (de Vries et al., 2019; Delobelle et al., 2020).

To perform NER with BERTje[8] and RobBERT[9], the Huggingface library with the Transformers architecture (pytorch-pretrained-bert) was used (Wolf et al., 2020). It provides the architecture of all 32+ pre-trained models in Natural Language Understanding and Natural Language Generation. From Huggingface Transformers[10] I used the run_ner.py script with the two different pre-trained models for generating evaluations and results. To do so, one should first install Tensorflow, Pytorch and Transformers. Within Transformers one can find 'examples' and 'pytorch' for the run_ner.py script. For the rest I applied standard settings.

3.2.2 Experimental Set-Up

I installed the Huggingface Transformers library to fine-tune the two Dutch Language Models on the 15th of May. In order to perform experiments I chose to differ on the number of training epochs. The number of epochs define the amount of times the complete data are passed through training. At first I fine-tuned on three epochs, after that on five epochs. Furthermore, the batch size was set on 8 and the seed was set on 777 to increase reproducibility.

3.3 Entity Linking

Entity Linking was performed after fine-tuning on BERT and retrieving predictions. In general Entity Linking is a difficult task because of the ambiguity of words and different entity mentions belonging to the same local geographical code.

To perform Entity Linking for this internship project prior steps had to be taken. The predictions of BERT only contain the actual predictions and not the tokens that belong to them. Therefore, the first step was to put both predictions and tokens back together. This way one has the tokens and the corresponding predictions. After that, I looked into whether there was a label present and if so which index inside of the sentence contained this label. After that, I performed string matching with the gazetteer. If the string inside the article and inside the gazetteer matched, the corresponding code or multiple codes of that place name or multiple place names was added to that article.

[8] https://github.com/wietsedv/bertje
[9] https://github.com/iPieter/RobBERT
[10] https://github.com/huggingface/transformers

4.Results

This chapter shows the results of this internship project. Section 4.1 explains results of fine-tuning BERTje and RobBERT on NER. Subsection 4.1.1 presents the results on the validation data and Subsection 4.1.2 shows results on the test data. Finally, Section 4.2 presents the results on Entity Linking.

4.1 Results Fine-Tuning BERT

For fine-tuning BERT on NER, evaluations are performed on the validation data to carry out experiments with the number of epochs and to select the number of epochs with the best results. The number of epochs with the best results is used to fine-tune BERT on the test data.

4.1.1 Results on Validation data

Model	Epochs	Label	Number	Precision	Recall	F-score	Macro Average
BERTje	3	GM	281	0.909	0.968	0.938	0.944
		PV	251	0.946	0.980	0.963	
		LD	9	0.9	1	0.947	
		CR	8	1	0.625	0.769	
	5	GM	281	0.916	0.972	0.943	0.941
		PV	251	0.949	0.980	0.965	
		LD	9	0.9	1	0.947	
		CR	8	1	0.625	0.769	
RobBERT	3	GM	281	0.873	0.928	0.9	0.773
		PV	251	0.897	0.972	0.933	
		LD	9	0.75	1	0.857	
		CR	8	0.571	0.5	0.533	
	5	GM	281	0.909	0.964	0.936	0.882
		PV	251	0.943	0.984	0.963	
		LD	9	0.818	1	0.9	
		CR	8	0.857	0.75	0.799	

Table 4: Results on Validation Data

Frequent vs. Infrequent Entities

Results of table Table 4 show the evaluations of fine-tuning on BERTje and RobBERT for NER on the validation data. The amount of labels in the validation data vary. The *gemeente (GM)* and *province (PV)* labels have a high number of labels, while the *landelijk* (LD) and *COROP (CR)* labels have a low number of labels.

BERTje and Number of Epochs

As shown in Table 4, precision and recall for fine-tuning BERTje on 3 epochs are high for all four categories. When fine-tuning BERTje on 5 epochs, precision of the *gemeente (GM)* and *provincie (PV)* label increases. For the *gemeente (GM)* label recall also increases when fine-tuning on 5 epochs. Furthermore, precision and recall are very high for the *landelijk (LD*) category while only precision is very high for the CR category.

RobBERT and Number of Epochs

As shown in Table 4, precision and recall for fine-tuning RobBERT on 3 epochs is high as well except for the *COROP(CR)* label. When fine-tuning on 5 epochs precision and recall of all four categories increase.

BERTje vs. RobBERT

As shown in Table 4, precision and recall are lower for RobBERT than for BERTje while fine-tuning on 3 epochs. Also, the LD and CR categories have lower precision. For the CR category recall also decreases for RobBERT in comparison to BERTje. However, when fine-tuning on 5 epochs precision and recall increases for RobBERT in comparison to fine-tuning on 3 epochs with RobBERT. Nevertheless, in comparison with BERTje when fine-tuning on 5 epochs, precision and recall for RobBERT are lower for all four categories except for recall of CR category.

Overall both systems have a high performance on NER for both 3 and 5 epochs. Which indicates that both systems are able to make accurate predictions on named entities. However, for both systems individually precision and recall increases while fine-tuning on 5 epochs. Therefore, 5 epochs are selected while fine-tuning BERTje and RobBERT on the test data. A more elaborate evaluation of the results can be found in Chapter 5.

4.1.2 Results on Test data

Model	Epochs	Label	Number	Precision	Recall	F-score	Macro Average
BERTje	5	PV	423	0.972	0.991	0.981	0.894
		GM	284	0.923	0.936	0.930	
		CR	44	0.894	0.955	0.923	
		LD	12	0.8	1	0.889	
RobBERT	5	PV	423	0.972	0.993	0.982	0.932
		GM	284	0.944	0.944	0.944	
		CR	44	0.896	0.977	0.935	
		LD	12	0.667	0.833	0.741	

Table 5: Results on Test Data

Frequent vs. Infrequent Entities

As shown in Table 5, the amount of labels for the test data is larger than the amount of labels for the validation data. As shown below, the amount of *COROP (CR)* labels increases significantly in comparison to the validation data. But the *landelijk (LD)* label still has a low number of labels.

Number of Epochs

The previous Section shows results on the validation data while doing experiments with the number of epochs. Five epochs has the highest performance scores for both BERTje and RobBERT. Five epochs is therefore used for fine-tuning on the test data.

BERTje

As illustrated in Table 5, precision and recall for BERTje of the *provincie (PV*) category increases in comparison to the precision and recall on the validation set in Table 4. Also precision for the *gemeente (GM)* category increases but recall decreases in comparison to the results of the validation data (Table 4). For BERTje there is also a shift for the *COROP (CR)* label, recall increases a lot in comparison to the results of the validation data. The *landelijk (LD)* label still contains a low number of annotations, but the precision decreases by 10% in comparison with the validation data.

RobBERT

For RobBERT one can see an improvement in performance on the test data presented in Table 5 compared to the results of the validation data as illustrated in Table 4. Precision increases for the *gemeente (GM)* label, but recall decreases in comparison to the validation data. For both the *provincie (PV)* as well as the *COROP (CR)* label both precision and recall increases compared to the validation data. But for the *landelijk (LD)* label, which contains a small number of annotation labels, precision and recall both decrease in comparison with the validation data. Overall performance for three labels increased in comparison to the validation data, only for the *gemeente (GM)* label recall decreased in comparison to the validation data.

BERTje vs. RobBERT

When comparing BERTje and RobBERT one can see that for the *provincie (PV)* label both models have a similar performance. However, there is a difference in performance on the *gemeente (GM)* label. Results show that RobBERT has higher performance on this label than BERTje. This is also the case for the *COROP (CR*) label as indicated in Table 5. However, whenever the amount of annotated labels decreases as for the *landelijk (LD)* label one can perceive that BERTje has higher performance than RobBERT.

4.2 Results on Named Entity Linking

To evaluate EL, difficult cases, such as long place names, or ambiguous place names are selected. The results in Table 6 show whether the entity mentioned is linked correctly to the corresponding code. I manually checked whether the code is correct, to gain insight in the performance of the Entity Linker.

	Sentence	**Actual Entity**	**Predicted Entity**	**Actual Code**	**Predicted Code**
1	In de provincie *Utrecht* is de grootste stijging zichtbaar.	Province	PV	PV26	PV26
2	In *Pijnacker Nootdorp* bedroeg de toename 81 procent.	Municipality (Gemeente)	GM	GM1926	GM1926
3	In de provincie *Groningen* is de daling van het aantal woningen waarvoor een bouwvergunning is verleend het sterkst.	Province	PV	PV20	PV20
4	In 2018 waren Súdwest Fryslân.	COROP	CR	CR05	-
4	De gemeente *Haarlemmerliede en Spaarnwoude*.	Municipality (Gemeente)	GM	GM0393	GM0393

5	Inwoners van *Den Haag, Almere* en *Amsterdam* hadden in 2010.	All three: Municipality (GM)	All three: GM	Amsterdam: GM0363 Almere: GM0034 Den Haag: GM0518	Amsterdam: GM0363 Almere: GM0034 Den Haag: -
6	In *Zuid Nederland*	Rural Area (Landelijk)	LD	LD04	LD04
7	*Groot Amsterdam* en *Groot Rijnmond*.	Both: COROP	Both: CR	Groot Amsterdam: CR23 Groot Rijnmond: CR29	Groot Amsterdam: CR23 Groot Rijnmond: CR29

Table 6: Performance of the Entity Linker

In sentence 1 in Table 6, Utrecht is stated as a province and the correct code is added to the document by the Entity Linker. Also for Pijnacker Nootdorp, the correct code is given by the Entity Linker. In sentence three Groningen can mean a municipality or a province. In context of sentence three it is about a province which is correctly predicted by the NER model. Also, again the Linker gives back the correct code as indicated in the gazetteer.

For sentence 4 the spelling of Friesland is in Frisian instead of Dutch, the prediction of the entity is correct but no code is given back. For sentence 5, again the correct code is given back by the Entity Linker. For sentence 6 for both Amsterdam as well as Almere the correct code is given back by the Linker. However, for Den Haag there is no code given back although the prediction is correct, but the formal spelling of Den Haag in the gazetteer used by the Entity Linker is ‘s-Gravenhage. For both sentence 7 and 8 the correct code is linked to the corresponding place names.

Altogether, results reveal that often the entity and corresponding code are correctly linked together. However, there are some difficult cases for which this is not the case. A more extensive evaluation of those results can be found in Chapter 5.

5. Discussion

The previous chapters have presented a method to create a rule-based baseline and a NER and EL system to increase findability of local geographical information. This chapter presents a discussion of these results in Section 5.1. Section 5.2 provides information of what improvements are still needed. In Section 5.3 suggestions for future work are described.

5.1 Method and Results

This section provides a description of the results of the rule-based system in Subsection 5.1.1 as a method to generate documents which included local geographical information. In Subsection 5.1.2 an explanation of the results on NER based on scientific literature will be set out. Then in Subsection 5.1.3 results on Named Entity linking will be discussed.

5.1.1 Rule-Based System

Difficult Cases

As indicated earlier, a rule-based system was used to retrieve documents that contained local geographical information. During manual annotations I created a gold standard of those articles. I looked into 1024 articles that were retrieved by the rule-based system to check whether those articles indeed included local geographical information. While annotating I came across multiple cases which were labelled as a region by the rule-based system, while in fact it was not a local geographical indication. Examples of such cases were Heel, Huizen, Waarde and Buren. All these examples could indicate a municipality but Heel could also indicate whole, Huizen could indicate houses, Waarde could indicate value and Buren could indicate neighbours.

While creating the rule-based system I decided to not change casing to overcome this problem. Whenever Huizen for example is mentioned lowercased, it was expected that it would not be retrieved as being a local geographical indication. However, Huizen as houses, could also be at the beginning of a sentence and therefore being capitalized. The same goes for the other examples and therefore manual annotations were important to perform.

Context

Furthermore, context is shown to be really important to take into account. Examples of this are:

(1) 80% van de studenten slaagden op de Universiteit Maastricht.
(2) Het onderzoek van mevrouw Bunschoten toonde dat aan.
(3) Het aantal vluchten vanaf Eindhoven Airport is met 60% gedaald sinds corona.

As in example (1) universities with a place name were also labelled as a region by the rule-based system. However, the previous word before 'Maastricht' gives a clue that Maastricht should not be identified as an entity. The same goes for surnames as in example (2), Bunschoten is a well-known surname in the Netherlands, but could also indicate a municipality. However, the previous word before the local geographical indication gives insight into whether an entity should be identified. Whereas in example (6) the system should look to the preceding word to get a clue whether it should be identified as an entity.

Overall, the results on the rule-based system show that all mentions are labelled as a region without taking context into account. Therefore, manual annotations are needed to create a gold standard of actual local geographical information. Overall context is shown to be really important to take into account. The BERT-based models are able to take both left and right context into account (Devlin et al., 2018) and are therefore selected for this internship project.

5.1.2 Results on NER

Results on Validation Data

Overall the performance of BERTje as well as RobBERT are high as shown in Chapter 4. However, one can see there are differences in performance. At first BERTje performs better than RobBERT with a lower amount of labels for the validation data. This result disagrees with the results presented by Delobelle et al. (2020) since those results show that RobBERT outperforms other language models and that less data is needed for the model to perform better. Delobelle et al. (2020) also state that RobBERT has more knowledge of Dutch because of different pre-training and a larger Dutch corpus. Taking the results of this internship project with regard to validation into account, this theory can however not be applied on the validation data.

Results on Test Data

On the other hand, RobBERT outperforms BERTje on the test data which consists of more labels. A possible reason for this difference between the results of performance of those two models could be that both BERTje and RobBERT used another pre-training regime (de Vries et al., 2019, Delobelle et al., 2020). BERTje was pre-trained on two objectives, as mentioned earlier, namely the MLM and SOP. While RobBERT was only pre-trained on the MLM. There was also a difference between the applied tokenizers. BERTje applies the BPE tokenizer of size 30K as explained in Chapter two, while the RobBERT v2, which is used for this project, contains an own created Dutch tokenizer as explained in Chapter two (Devlin et al., 2018, Delobelle et al., 2020). Another reason for better performance of RobBERT is that RobBERT was pre-trained on three extra corpora in comparison to BERTje.

Validation Data vs. Test Data

The *landelijk (LD)* category contains a lower amount of labels for both the test data as well as for the validation data. From Table 4 and Table 5 it is shown that RobBERT performs better on the validation data than on the test data for the *landelijk (LD)* label. While Delobelle et al. (2020) stated that RobBERT does not need much data to outperform other models. However, this difference for the *landelijk (LD)* label is negligible, because of the low amount of labels it seems like a big difference of performance while in fact RobBERT only makes one extra prediction wrong.

Altogether, the results show that RobBERT outperforms BERTje on this NER task when using 5 epochs and a larger amount of labels from the test data is used. Because of this bigger amount of labels one can conclude that RobBERT is better at generalizing new data.

5.1.3 Results on EL

The results presented in Chapter 4 on EL show that overall the Entity Linker links the correct code to the correct local geographical indication. However, when a different spelling occurs such as 'Noardeast Fryslân', the Entity Linker does not link any code to this local geographical indication. This is due to the fact that the spelling is in Frisian and not in standard Dutch. Also, there is variation on spelling of for example Den Haag which is indicated in the gazetteer as 's-Gravenhage. In standard written and spoken Dutch however, 's-Gravenhage is often indicated as Den Haag and therefore the articles cannot be linked properly.

5.2 Recommendations and Future work

Overall, the ambiguous words from the rule-based system are labelled as not being a local geographical indication during annotations. Also, both models show a high performance. However, to get a better understanding on which model performs best I would suggest creating more annotations. As one can see for the categories with a lower amount of labels, the predictions vary. Furthermore, different spellings for local geographical indications should be added to the Entity Linker. This will cause the Entity Linker to perform better on cases where for example Frisian was used instead of Dutch. Another point of improvement would be to develop an evaluation set for Entity Linking. The above mentioned improvements could be implemented in future work.

6. Conclusion

This chapter provides the conclusion of this internship project. The aim of this project was to find out whether NER and EL increase findability of local geographical information in the CerBeruS search engine. The research question was:

Will findability of local geographical data in the CerBerus Search increase when applying Named Entity Recognition in combination with Named Entity Linking?

To find out whether NER and Named Entity improved findability of local geographical data I fine-tuned BERT on NER and applied EL. The results of the NER systems of this project show that both BERTje as well as RobBERT have high performance on predicting the correct local geographical indication inside a text. This indicates that our NER system gives strong results, as BERT models are able to take context into account, which TF-IDF cannot. There was no evaluation set available to compare the method of this internship project with the previous method but I expect a BERT-based model to be more precise because of the context that is taken into account. For EL, I manually checked the codes that the system gives back. Results show that combining our NER system and EL system gives back the correct local geographical code. Overall Statistics Netherlands is very enthusiastic about the high performance of the NER system in combination with EL and will implement the system into the CerBeruS Search Engine.

Contribution

As indicated in Chapter 2 lack of domain specific annotations is a huge challenge for NER systems. But the annotated data for this internship project is domain specific and is therefore beneficial for Statistics Netherlands. These annotations can now be used to train any kind of NER system for local geographical data for Statistics Netherlands.

Also, the code used for this internship project is of great use for other Named Entity Tasks by Statistics Netherland. The README (Appendix IV) serves as a guideline on how to use the code for the local geographical indications, but it can also serve as a guideline to retrieve new Named Entities and link them with corresponding codes. However, if Statistics Netherlands wants to create a NER system on new entities, new manual annotations should be performed. But with the rule-based system it is easier to retrieve only those documents that contain the Named Entities, Statistics Netherlands is searching for and therefore, annotation time decreases.

Recommendations

Based on the results I would recommend to use RobBERT, since performance of this model is the highest on a larger amount of labels. Which indicates that RobBERT is able to generalize better on new

data. This system performs well on its own and is able to retrieve the Named Entities from texts by taking the context into account. As stated earlier context is really important and the previous approach of Statistics Netherlands did not apply this. Therefore, by using this bidirectional model I expect a better findability of the local geographical entities. On top of that, the combination of the NER systems and EL systems shows high performance, however I would first recommend developing an evaluation set for Entity Linking. Furthermore, I would recommend to expand the gazetteer for different spellings. For example adding 'Den Haag' and 'Den Bosch' and also add translations of Fries to the gazetteer. In this way Entity Linking will gain better performance as well.

Conclusion

To conclude, the aim of this internship has been achieved meaning that findability of local geographical information increases when using NER and EL. This goal was achieved by creating the rule-based baseline to retrieve documents containing local geographical information. Through manual annotating 1024 documents I was able to fine-tune BERTje and RobBERT on NER. The results show that both systems have high performance and are able to predict local geographical indications in texts. In combination with EL, disambiguation was possible for difficult cases. The results show that this combined system performs well and makes local geographical entity extraction, thus findability, more accurate.

References

Agarwal, B., & Mittal, N. (2014). Text classification using machine learning methods-a survey. In *Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28-30, 2012* (pp. 701-709). Springer, New Delhi.

Agerri, R., & Rigau, G. (2016). Robust multilingual named entity recognition with shallow semi-supervised features. *Artificial Intelligence*, *238*, 63-82.

Allein, L., Leeuwenberg, A., & Moens, M. F. (2020). Binary and multitask classification model for Dutch anaphora resolution: Die/dat prediction. *arXiv preprint arXiv:2001.02943*.

Akbik, A., Blythe, D., & Vollgraf, R. (2018, August). Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics* (pp. 1638-1649).

Akbik, A., Bergmann, T., & Vollgraf, R. (2019, June). Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 724-728).

Baevski, A., Edunov, S., Liu, Y., Zettlemoyer, L., & Auli, M. (2019). Cloze-driven pretraining of self-attention networks. *arXiv preprint arXiv:1903.07785*.

Carreras, X., Márquez, L., & Padró, L. (2002). Named Entity Extraction using AdaBoost, proceeding of the 6th Conference on Natural language learning. In *Association for Computational Linguistics*.

Centraal Bureau voor de Statistiek. (2004, 8 juni). *Accounting for Sustainable Development*. https://www.cbs.nl/nl-nl/dossier/nederland-regionaal/gemeente/gemeenten-en-regionale-indelingen/landelijk-dekkende-indelingen

Centraal Bureau voor de Statistiek. (2018, 22 mei). *Werkzame bevolking Noordoost-Noord-Brabant, onderwijs*. https://www.cbs.nl/nl-nl/maatwerk/2018/21/werkzame-bevolking-noordoost-noord-brabant-onderwijs

Centraal Bureau voor de Statistiek. (2018, 5 september). *Huizenprijzen op niveau van voor de kredietcrisis.* https://www.cbs.nl/nl-nl/nieuws/2018/36/huizenprijzen-op-niveau-van-voor-de-kredietcrisis

Centraal Bureau voor de Statistiek. (2019, 6 november). *Goederenhandel Zuidoost-Noord-Brabant*. https://www.cbs.nl/nl-nl/maatwerk/2019/45/goederenhandel-zuidoost-noord-brabant

Centraal Bureau voor de Statistiek. (2021, 17 februari). *Landelijk dekkende indelingen*. https://www.cbs.nl/nl-nl/dossier/nederland-regionaal/gemeente/gemeenten-en-regionale-indelingen/landelijk-dekkende-indelingen

Collobert, R., & Weston, J. (2008, July). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning* (pp. 160-167).

Delobelle, P., Winters, T., & Berendt, B. (2020). Robbert: a dutch roberta-based language model. *arXiv preprint arXiv:2001.06286*.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Feldman, R., & Dagan, I. (1995). Knowledge Discovery in Textual Databases (KDT). *KDD-95 Proceedings*. Published.

Florian, R., Ittycheriah, A., Jing, H., & Zhang, T. (2003). Named entity recognition through classifier combination. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003* (pp. 168-171).

Gayathri, K., & Marimuthu, A. (2013, January). Text document pre-processing with the KNN for classification using the SVM. In *2013 7th International Conference on Intelligent Systems and Control (ISCO)* (pp. 453-457). IEEE.

Gillick, D., Brunk, C., Vinyals, O., & Subramanya, A. (2015). Multilingual language processing from bytes. *arXiv preprint arXiv:1512.00103*.

Goyal, A., Gupta, V., & Kumar, M. (2018). Recent named entity recognition and classification techniques: a systematic review. *Computer Science Review*, *29*, 21-43.

Gokaslan, A., & Cohen, V. (2019). Openwebtext corpus. *urlhttp://Skylion007. github. io/OpenWebTextCorpus*.

Grisham, R., & Sundheim, B. (1996). Message Understanding: a brief history. In *Proceedings of the Sixth Message Understanding Conference*.

Habibi, M., Weber, L., Neves, M., Wiegandt, D. L., & Leser, U. (2017). Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, *33*(14), i37-i48.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., ... & Weikum, G. (2011, July). Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (pp. 782-792).

Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Kaur, G. (2014). Usage of regular expressions in NLP. *International Journal of Research in Engineering and Technology IJERT*, *3*(01), 7.

Klie, J.-C., Bugert, M., Boullosa, B., Eckart de Castilho, R. and Gurevych, I. (2018): The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation. *In Proceedings of System Demonstrations of the 27th International Conference on Computational Linguistics (COLING 2018), Santa Fe, New Mexico, USA*

Koehn, P. (2005, September). Europarl: A parallel corpus for statistical machine translation. In *MT summit* (Vol. 5, pp. 79-86).

Kuru, O., Can, O. A., & Yuret, D. (2016, December). Charner: Character-level named entity recognition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 911-921).

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Le, H., Vial, L., Frej, J., Segonne, V., Coavoux, M., Lecouteux, B., ... & Schwab, D. (2019). Flaubert: Unsupervised language model pre-training for french. *arXiv preprint arXiv:1912.05372*.

Li, B., Zhao, Z., Liu, T., Wang, P., & Du, X. (2016, December). Weighted neural bag-of-n-grams model: New baselines for text classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (pp. 1591-1600).

Liang, C., Yu, Y., Jiang, H., Er, S., Wang, R., Zhao, T., & Zhang, C. (2020, August). Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 1054-1064).

Liu, S., Tang, B., Chen, Q., & Wang, X. (2015). Effects of semantic features on machine learning-based drug name recognition systems: word embeddings vs. manually constructed dictionaries. *Information*, *6*(4), 848-865.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Luoma, J., & Pyysalo, S. (2020). Exploring Cross-sentence Contexts for Named Entity Recognition with BERT. *arXiv preprint arXiv:2006.01563*.

Mac Kim, S., & Cassidy, S. (2015, December). Finding names in trove: named entity recognition for Australian historical newspapers. In *Proceedings of the Australasian Language Technology Association Workshop 2015* (pp. 57-65).

Martin, L., Muller, B., Suárez, P. J. O., Dupont, Y., Romary, L., de la Clergerie, É. V., ... & Sagot, B. (2019). Camembert: a tasty french language model. *arXiv preprint arXiv:1911.03894*.

Martins, P. H., Marinho, Z., & Martins, A. F. (2019). Joint learning of named entity recognition and entity linking. *arXiv preprint arXiv:1907.08243*.

Mikolov, T., Yih, W. T., & Zweig, G. (2013, June). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 746-751).

Nguyen, D. Q., Billingsley, R., Du, L., & Johnson, M. (2015). Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics*, *3*, 299-313.

Paik, J. H. (2013, July). A novel TF-IDF weighting scheme for effective ranking. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval* (pp. 343-352).

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Polignano, M., Basile, P., De Gemmis, M., Semeraro, G., & Basile, V. (2019). Alberto: Italian BERT language understanding model for NLP challenging tasks based on tweets. In *6th Italian Conference on Computational Linguistics, CLiC-it 2019* (Vol. 2481, pp. 1-6). CEUR.

Ramos, J. (2003, December). Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning* (Vol. 242, No. 1, pp. 29-48).

Ritter, A., Clark, S., & Etzioni, O. (2011, July). Named entity recognition in tweets: an experimental study. In *Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 1524-1534).

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, *24*(5), 513-523.

Sang, E. F., & De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Sang, E. F. (2002). Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2002* (pp. 155-158).

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, *34*(1), 1-47.

Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Socher, R., Bauer, J., Manning, C. D., & Ng, A. Y. (2013, August). Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 455-465).

Team, T. I. (z.d.). *INCEpTION User Guide*. Retrieved on 7th of May, 2021, from https://inception-project.github.io/releases/0.19.3/docs/user-guide.html#_introduction

Trinh, T. H., & Le, Q. V. (2018). A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Virtanen, A., Kanerva, J., Ilo, R., Luoma, J., Luotolahti, J., Salakoski, T., ... & Pyysalo, S. (2019). Multilingual is not enough: BERT for Finnish. *arXiv preprint arXiv:1912.07076*.

de Vries, W., van Cranenburgh, A., Bisazza, A., Caselli, T., van Noord, G., & Nissim, M. (2019). Bertje: A dutch bert model. *arXiv preprint arXiv:1912.09582*.

Wolf, T., Chaumond, J., Debut, L., Sanh, V., Delangue, C., Moi, A., ... & Rush, A. M. (2020, October). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 38-45).

Wu, S., & Dredze, M. (2019). Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. *arXiv preprint arXiv:1904.09077*.

Yadav, V., & Bethard, S. (2019). A survey on recent advances in named entity recognition from deep learning models. *arXiv preprint arXiv:1910.11470*.

Yamada, I., Takeda, H., & Takefuji, Y. (2015, July). Enhancing named entity recognition in twitter messages using entity linking. In *Proceedings of the Workshop on Noisy User-generated Text* (pp. 136-140).

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Yun-tao, Z., Ling, G., & Yong-cheng, W. (2005). An improved TF-IDF approach for text classification. *Journal of Zhejiang University-Science A*, *6*(1), 49-55.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision* (pp. 19-27).

Appendices

In this chapter the appendices of this internship project are presented. Appendix I provides an example of an article used during annotations. Appendix II shows an example of the output of the rule-based system. Appendix III presents the annotation guidelines and the gazetteer that is used for this internship project. Finally Appendix IV provides the README for the code used during this internship project.

Appendix I

Bloemendaal was ook in 2005 de rijkste gemeente van Nederland. De inwoners van de gemeenten in Noord Holland zijn nog altijd relatief rijk en die in Groningen en Friesland naar verhouding arm. Bloemendaal en Reiderland, twee uitersten Na Bloemendaal volgden in de top Wassenaar, Blaricum en Abcoude. Zes van de tien gemeenten met het hoogste inkomen lagen in Noord Holland. De inwoners in de Groningse gemeenten Reiderland, Eemsmond en Pekela bevonden zich onderaan de inkomensladder. Enschede en Emmen onderaan Ook in de meeste steden met meer dan 100 duizend inwoners week het gemiddeld inkomen maar weinig af van het landelijk gemiddelde. Uitschieters naar beneden waren Enschede en Emmen. De inwoners van Haarlemmermeer hadden het hoogste inkomen. Van de gemeenten met 50 tot 100 duizend inwoners lag het gemiddeld inkomen in Amstelveen 25 procent boven het gemiddelde.

Appendix II

Table 1 below shows the output of the rule-based system. Column 1 shows the sentence in the text in which a placename is stated. Column 2 shows the tokens inside the sentence, column 3 the actual place name. In column 4 the part-of-speech is added and finally in column 5 the predicted label is stated.

0	Bloemendaal	Bloemendaal	PROPN	REGIO
0	was		AUX	O
0	ook		ADV	O
0	in		ADP	O
0	2005		NUM	O
0	de		DET	O
0	rijkste		ADJ	O
0	gemeente		NOUN	O
0	van		ADP	O
0	Nederland		PROPN	O
0	.		PUNCT	O
3	De		DET	O
3	inwoners		NOUN	O
3	van		ADP	O
3	de		DET	O
3	gemeenten		NOUN	O
3	in		ADP	O
3	Noord	Noord Holland	PROPN	REGIO
3	Holland	Noord Holland	PROPN	REGIO
3	zijn		AUX	O
3	nog		ADV	O
3	altijd		ADV	O
3	relatief		ADJ	O
3	rijk		ADJ	O
3	en		CCONJ	O
3	die		PRON	O
3	in		ADP	O
3	Groningen	Groningen	PROPN	REGIO
3	en		CCONJ	O
3	Friesland	Friesland	PROPN	REGIO
3	naar		ADP	O
3	verhouding		NOUN	O
3	arm		NOUN	O
3	.		PUNCT	O

Table 1: Rule-Based system output

Appendix III

Annotations Guidelines

- BIO schema used automatically with Inception. Example: Noord Holland with B-PV and I-PV
- Labels that were used where retrieved from the gazetteer provided by Statistics Netherlands and consisted out of:
    - LD (national i.e. landelijk) (Table 1)
    - PV (province) (Table 2)
    - CR (regional area i.e. COROP) (Table 3)
    - GM (municipality i.e. gemeente). (Table 4)
- To know what label to apply four different tables were used. Table 1 for Rural Areas, table 2 for Provinces, table 3 for COROP's and table 4 for municipalities. Most of the time one knows whether something is a local geographical indication, but when there was doubt this table was used.
- During annotating there were some difficult cases which should not be annotated as local geographical indications. Those difficult cases are shown in table 5.
- When an University, airport or surname was mentioned it was never annotated as local geographical indication.
- 's-Gravenhage (Den-Haag) and 's-Hertogenbosch (Den-Bosch) were annotated as local geographical indication. Since most often in the articles it was Den-Haag and Den-Bosch difficulties could arise with the Entity Linking.
- Also Frisian spelling is taken into account as shown in table Table 2 and Table 3.
- For the ambiguous words it is important for the annotator to take context of previous and preceding token into account. Always whenever it is about a municipality or a province this is clearly indicated. For example: 'In de gemeente Heel waren er weinig inbraken afgelopen jaar'. Which means: 'In the municipality Heel there where little robberies last year'. From this context the annotator should observe it is indeed about a municipality.

LD Label		
Noord Nederland	West Nederland	Zuidwest Nederland
Oost Nederland	Zuid Nederland	

Table 1: Annotations Scheme for LD Label

PV Label			
Groningen	Flevoland	Zuid Holland	Overijssel
Fryslan (Friesland)	Gelderland	Zeeland	Noord Holland
Drenthe	Utrecht	Noord Brabant	Limburg

Table 2: Annotation Scheme for PV Label

CR Label					
Oost Groningen	Zuidwest Friesland (Súdwest – Fryslân)	Zuidwest Drenthe	Veluwe	Utrecht	Agglomeratie Haarlem
Delfzijl en omgeving	Zuidoost Friesland (Súdeast Fryslân)	Noord Overijssel	Achterhoek	Kop van Noord Holland	Zaanstreek
Overig Groningen	Noord Drenthe	Zuidwest Overijssel	Arnhem/Nijmegen	Alkmaar en omgeving	Groot Amsterdam

Noord Friesland (Noard Fryslân)	Zuidoost Drenthe	Twente	Zuidwest Gelderland	IJmond	Het Gooi en Vechtstreek
Agglomeratie Leiden en Bollenstreek	Groot Rijnmond	West Noord Brabant	Noord Limburg	Oost Zuid Holland	Overig Zeeland
Agglomeratie ‘s Gravenhage	Zuidoost Zuid Holland	Noordoost Noord Brabant	Midden Limburg	Zuidoost Noord Brabant	Zuid Limburg
Delft en Westland	Zeeuws Vlaanderen	Midden Noord Brabant	Zuid Limburg		

Table 3: Annotation Scheme for CR Label

GM Label							
Aa en Hunze	Alphen en Riel	Asten	Bellingwedde	Bierum	Borne	Genderen	Gulpen
Aagtekerke	Alphen-Chaam	Avenhorn	Bellingwolde	Biervliet	Borsele	Gendringen	Gulpen-Wittem
Aalburg	Altena	Avereest	Bemelen	Biesland	Borssele	Gendt	Haaften
Aalsmeer	Ambt Delden	Axel	Bemmel	Biggekerke	Boschkapelle	Genemuiden	Haaksbergen
Aalst	Ambt Montfort	Baarderadeel	Bennebroek	Bijlmermeer	Boskoop	Gennep	Haamstede
Aalten	Ambt-Almelo	Baardwijk	Benschop	het Bildt	Bovenkarspel	Gerverscop	Haaren
Ter Aar	Ambt-Doetinchem	Baarland	Benthorn	De Bilt	Boxmeer	Gestel en Blaarthem	Haarlem
Aardenburg	Ambt-Hardenberg	Baarle-Nassau	Benthuizen	Bingelrade	Boxtel	Geulle	Haarlemmerliede
Aarlanderveen	Ambt-Ommen	Baarn	Berg en Dal	Binnenmaas	Brakel	Giessen	Haarlemmerliede en Spaarnwoude
Aarle-Rixtel	Ambt-Vollenhove	Baexem	Berg en Terblijt	Bladel	Brandwijk	Giessenburg	Haarlemmermeer
Abbekerk	Amby	Baflo	Bergambacht	Bladel en Netersel	Breda	Giessendam	Haarzuilens
Abbenbroek	Ameide	Bakel en Milheeze	Bergeijk	Blankenham	Brederwiede	Giessenlanden	Haastrecht

Abcoude	Ameland	Balgoij	Bergen (L.)	Blaricum	Breskens	Giessen-Nieuwkerk	Haelen
Abcoude-Baambrugge	Amerongen	Bangert	Bergen (NH.)	Bleiswijk	Breukelen	Gieten	Hagestein
Abcoude-Proosdij	Amersfoort	Barendrecht	Bergen op Zoom	Bleskensgraaf	Breukelen-Nijenrode	Giethoorn	Halderberge
Abtsregt	Ammerstol	Barneveld	Bergeyk	Bleskensgraaf en Hofwege	Breukelen-Sint Pieters	Gilze en Rijen	Halsteren
Achthoven	Ammerzoden	Barradeel	Bergh	Bloemendaal	Brielle	Ginneken en Bavel	Den Ham
Achtkarspelen	Amstelveen	Barsingerhorn	Bergharen	Blokker	Broek	Goedereede	Haps
Achttienhoven (U.)	Amstenrade	Barwoutswaarder	Berghem	Blokzijl	Broek in Waterland	Goeree-Overflakkee	Hardenberg
Achttienhoven (ZH.)	Amsterdam	Batenburg	Bergschenhoek	Boarnsterhim	Broek op Langedijk	Goes	Harderwijk
Ackersdijk en Vrouwenregt	Andel	Bath	Berkel en Rodenrijs	Bocholtz	Broekhuizen	Goidschalxoord	Hardinxveld
Adorp	Andijk	Bathmen	Berkel-Enschot	Bodegraven	Broeksittard	Goirle	Hardinxveld-Giessendam
Aduard	Angerlo	Bedum	Berkelland	Bodegraven-Reeuwijk	Bronckhorst	Gooise Meren	Haren
Aengwirden	Ankeveen	Beegden	Berkenrode	Boekel	Brouwershaven	Goor	Harenkarspel
Akersloot	Anloo	Beek (L.)	Berkenwoude	Ten Boer	Bruinisse	Gorinchem	Harlingen
Alblasserdam	Anna Paulowna	Beek (NB.)	Berkhout	Bokhoven	Brummen	Gorssel	Harmelen
Albrandswaard	Apeldoorn	Beek en Donk	Berlicum	Bolsward	Brunssum	Gouda	Haskerland
Albrandswaard (oud)	Appeltern	Beekdaelen	Bernheze	Den Bommel	Budel	Gouderak	Hasselt
Alem, Maren en Kessel	Appingedam	Beemster	Bernisse	Bommenede	Buggenum	Goudriaan	Hattem
Alkemade	Arcen en Velden	Beers	Besoijen	Boornsterhem	Buiksloot	Goudswaard	Havelte
Alkmaar	Arkel	Beerta	Best	Borculo	Bunde	Graafstroom	Hazerswoude
Almelo	Arnemuiden	Beesd	Beugen en Rijkevoort	Borger	Bunnik	Graauw en Langendam	Hedel
Almere	Arnhem	Beesel	Beuningen	Borger-Odoorn	Bunschoten	Grafhorst	Hedikhuizen
Almkerk	Asperen	Beets	Beusichem	Borgharen	Buren	Graft	Heel
Alphen	Assen	Beilen	Beverwijk	Borkel en Schaft	Burgh	Graft-De Rijp	Heel en Panheel
Alphen aan den Rijn	Assendelft	Belfeld	Biert	Born	Bussum	Gramsbergen	Heemskerk
Buurmalsen	Deil	Dommelen	Duiven	Eijgelshoven	Everdingen	Grathem	Heemstede
Cabouw	Delfshaven	Dongen	Duivendijke	Eijsden	Ewijk	Grave	Heenvliet
Cadier en Keer	Delft	Dongeradeel	Duizel en Steensel	Eijsden-Margraten	Ezinge	's-Graveland	Heer
Cadzand	Delfzijl	Doniawerstal	Den Dungen	Eindhoven	Ferwerderadeel	's-Gravenambacht	's-Heer Hendrikskinderen
Callantsoog	Denekamp	Doorn	Dussen	Elburg	Ferwerderadiel	's-Gravendeel	Heer Oudelands Ambacht

Capelle	Deurne	Doornspijk	Dussen, Munster en Muilkerk	Elkerzee	Fijnaart en Heijningen	's-Gravenhage (gemeente) (Den Haag)	's-Heer-Abtskerke
Capelle aan den IJssel	Deurne en Liessel	Doorwerth	Dwingeloo	Ellemeet	Finsterwolde	's-Gravenmoer	's-Heer-Arendskerke
Castricum	Deursen en Dennenburg	Dordrecht	Echt	Ellewoutsdijk	Franeker	's-Gravenpolder	Heerde
Chaam	Deventer	Dorth	Echteld	Elsloo	Franekeradeel	's-Gravenzande	's-Heerenhoek
Charlois	Didam	Drechterland	Echt-Susteren	Elst	De Friese Meren	Gravesloot	Heerenveen
Cillaarshoek	Dieden, Demen en Langel	Dreischor	Edam	Elten	De Fryske Marren	Grevenbicht	Heerewaarden
Clinge	Diemen	Dreumel	Edam-Volendam	Emmen	Gaasterland	Grijpskerk	Heerhugowaard
Coevorden	Diepenheim	Driebergen	Ede	Emmikhoven	Gaasterlân-Sleat	Grijpskerke	Heerjansdam
Colijnsplaat	Diepenveen	Driebergen-Rijsenburg	Eede	Empel en Meerwijk	Gameren	Groede	Heerlen
Cothen	Diessen	Driebruggen	Eelde	Engelen	Gapinge	Groeneveld	Heesch
Cranendonck	Diever	Driel	Eemnes	Enkhuizen	Gassel	Groenlo	Heeswijk
Cromstrijen	Dinkelland	Driewegen	Eemsdelta	Enschede	Gasselte	Groesbeek	Heeswijk-Dinther
Cromvoirt	Dinteloord en Prinsenland	Drimmelen	Eemsmond	Epe	Geertruidenberg	Groet	Heeze
Cuijk	Dinther	Drongelen	Eenrum	Ermelo	Geervliet	Groningen (gemeente)	Heeze-Leende
Cuijk en Sint Agatha	Dinxperlo	Drongelen, Haagoord, Gansoyen, Doevere	Eersel	Erp	Geffen	Gronsveld	Hefshuizen
Culemborg	Dirksland	Dronten	Eethen	Esch	Geldermalsen	Groot-Ammers	Hei- en Boeicop
Dalen	Dodewaard	Drunen	Eethen, Genderen en Heesbeen	Escharen	Geldrop	Groote Lindt	Heille
Dalfsen	Doesburg	Druten	Egmond	Est en Opijnen	Geldrop-Mierlo	Grootebroek	Heiloo
Dantumadeel	Doetinchem	Dubbeldam	Egmond aan Zee	Etersheim	Geleen	Grootegast	Heinenoord
Dantumadiel	Dokkum	Duist	Egmond-Binnen	Etten en Leur	Gemert	Grosthuizen	Heinkenszand
Darthuizen	Domburg	Duiveland	Eibergen	Etten-Leur	Gemert-Bakel	Grubbenvorst	Heino
Hekelingen	Het Hogeland	IJzendijke	Korendijk	Leusden	Maasgouw	Middenschouwen	Nieuwe Pekela
Hekendorp	Hollands Kroon	IJzendoorn	Kortenhoef	Lexmond	Maashees en Overloon	Middenveld	Nieuwe Tonge
Helden	Holten	Ilpendam	Kortgene	Lichtenvoorde	Maasland	Midwolda	Nieuwegein
Den Helder	Hontenisse	Itteren	Koudekerk	Liemeer	Maasniel	Midwoud	Nieuwendam
Hellendoorn	Hoofdplaat	Ittervoort	Koudekerk aan den Rijn	Liempde	Maassluis	Mierlo	Nieuwenhagen

Hellevoetsluis	Hoog en Woud Harnasch	Jaarsveld	Koudekerke	Lienden	Maastricht	Mijdrecht	Nieuwenhoorn
Helmond	Hoogbloklandd	Jabeek	Krabbendijke	De Lier	Made	De Mijl	Nieuwe-Niedorp
Helvoirt	Hooge en Lage Mierde	Jacobswoude	Kralingen	Lierop	Made en Drimmelen	Mijnsheerenland	Nieuwer-Amstel
Hemelumer Oldeferd	Hooge en Lage Zwaluwe	Jisp	Krimpen aan de Lek	Lieshout	Margraten	Mill en Sint Hubert	Nieuwerkerk
Hemelumer Oldephaerd en Noordwolde	Hoogeloon, Hapert en Casteren	Jutphaas	Krimpen aan den IJssel	Liesveld	Mariekerke	Millingen	Nieuwerkerk aan den IJssel
Hemmen	Hoogeveen	Kaag en Braassem	Krimpenerwaard	Limbricht	Markelo	Millingen aan de Rijn	Nieuweschans
Hendrik-Ido-Ambacht	Hoogeveen in Delfland	Kalslagen	Krommenie	Limmen	Marken	Moerdijk	Nieuw-Ginneken
Hengelo (Gld.)	Hoogeveen in Rijnland	Kamerik	Kruiningen	Linden	De Marne	Moergestel	Nieuw-Helvoet
Hengelo (O.)	Hoogezand	Kamerik Houtdijken	Kuinre	Lingewaal	Marum	Moerhuizen	Nieuwkoop
Hengstdijk	Hoogezand-Sappemeer	Kamerik Mijzijde	Kwadijk	Lingewaard	Maurik	Moerkapelle	Nieuwkuijk
Hennaarderadeel	Hoogkarspel	Kampen	Laagbloklandd	Linne	Medemblik	Molenaarsgraaf	Nieuwland
Hensbroek	Hoogkerk	Kamperveen	Laag-Nieuwkoop	Linschoten	Meeden	Molenlanden	Nieuwland, Kortland en 's-Graveland
Herkingen	Hoogland	Kantens	Laarbeek	Lisse	Meerdervoort	Molenwaard	Nieuw-Lekkerland
Herpen	Hoogmade	Kapelle	Landerd	Lith	Meerkerk	Monnickendam	Nieuwleusen
Herpt	Hoogvliet	Katendrecht	Landgraaf	Lithoijen	Meerlo	Monster	Nieuwolda
Herten	Hoogwoud	Kats	Landsmeer	Littenseradeel	Meerlo-Wanssum	Montferland	Nieuwpoort
's-Hertogenbosch (Den Bosch)	Hoorn	Kattendijke	Langbroek	Littenseradiel	Meerssen	Montfoort	Nieuwstadt
Herwen en Aerdt	Hoornaar	Katwijk	Lange Ruige Weide	Lochem	Meeuwen	Montfort	Nieuwveen
Herwijnen	Horn	Katwoude	Langedijk	Loenen	Meeuwen, Heel en Babyloniënbroek	Mook en Middelaar	Nieuwveen in Delfland
Heteren	Horssen	Kedichem	Langerak	Loenen en Wolveren	Megen, Haren en Macharen	Moordrecht	Nieuwvliet
Heukelum	Horst	Kerkrade	Lansingerland	Loenersloot	Meierijstad	Muiden	Nieuw-Vossemeer
Heumen	Horst aan de Maas	Kerkwerve	Laren (Gld.)	Lonneker	Meijel	Munstergeleen	Nigtevecht
Heusden	Houten	Kerkwijk	Laren (NH.)	Loon op Zand	Melick en Herkenbosch	Muntendam	Nijefurd
Heythuysen	Houthem	Kessel	Leek	Loosdrecht	Meliskerke	Naaldwijk	Nijeveen
Hillegersberg	Houtrijk en Polanen	Kesteren	Leende	Loosduinen	Melissant	Naarden	Nijkerk

Hillegom	Huijbergen	Kethel en Spaland	Leens	Lopik	Menaldumadeel	Naters	Nijmegen
Hilvarenbeek	Huisseling en Neerloon	Kijfhoek	Leerbroek	Loppersum	Menameradiel	Neder-Betuwe	Nisse
Hilversum	Huissen	Klaaswaal	Leerdam	Losser	Menterwolde	Nederhemert	Nissewaard
Hindeloopen	Huizen	Kleine Lindt	Leersum	Luyksgestel	Meppel	Nederhorst den Berg	Nistelrode
Hodenpijl	Hulsberg	Kleverskerke	Leeuwarden	Maarheeze	Merkelbeek	Nederlek	Noardeast-Fryslân
Hoedekenskerke	Hulst	Klimmen	Leeuwarderadeel	Maarn	Mesch	Nederslingelandt	Noorbeek
Hoek	Hummelo en Keppel	Kloetinge	Leiden	Maarssen	Mheer	Nederweert	Noord-Beveland
Hoeksche Waard	Hunsel	Kloosterburen	Leiderdorp	Maarssenbroek	Middelburg (Z.)	Neede	Noordbroek
Hoenkoop	Hurwenen	Klundert	Leidschendam	Maarsseveen	Middelburg (ZH.)	Neer	Noorddijk
Hoensbroek	Idaarderadeel	Kockengen	Leidschendam-Voorburg	Maartensdijk	Middelharnis	Neerijnen	Noordeloos
Hoevelaken	IJlst	Koedijk	Leimuiden	Maasbracht	Middelie	Neeritter	Noordenveld
Hoeven	IJsselham	Koewacht	Lekkerkerk	Maasbree	Middelstum	Nibbixwoud	Noorder-Koggenland
Hof van Delft	IJsselmonde	Koggenland	Lelystad	Maasdam	Midden-Delfland	Niedorp	Noordgouwe
Hof van Twente	IJsselmuiden	Kollumerland en Nieuwkruisland	Lemsterland	Maasdonk	Midden-Drenthe	Nieuw- en Sint Joosland	De Noordoostelijke Polder
Hofwegen	IJsselstein	Koog aan de Zaan	Leudal	Maasdriel	Midden-Groningen	Nieuw-Beijerland	Noordoostpolder
Noord-Polsbroek	Oldebroek	Ooststellingwerf	Oudenhoorn	Peel en Maas	Reek	Rijsoort	Rucphen
Noord-Scharwoude	Oldehove	Oostvoorne	Oude-Niedorp	Peize	Reeuwijk	Rijsoort en Strevelshoek	Ruijven
Noord-Waddinxveen	Oldekerk	Oostzaan	Oudenrijn	Pekela	Reiderland	Rijssen	Ruinen
Noordwelle	Oldemarkt	Ootmarsum	Ouder-Amstel	Pernis	Reimerswaal	Rijssen-Holten	Ruinerwold
Noordwijk	Oldenzaal	Openbaar Lichaam Z.IJ.P.	Ouderkerk	Petten	Renesse	Rijswijk (NB.)	Ruurlo
Noordwijkerhout	Olst	Ophemert	Ouderkerk aan den IJssel	Peursum	Renkum	Rijswijk (ZH.)	Ruwiel
Nootdorp	Olst-Wijhe	Oploo, Sint Anthonis en Ledeacker	Oude-Tonge	Philippine	Renswoude	Rilland	Sambeek
Norg	Ommen	Opmeer	Oudewater	Piershil	Retranchement	Rilland-Bath	Sandelinge-Ambacht
Nuenen, Gerwen en Nederwetten	Onderbanken	Opperdoes	Oudheusden	Pijnacker	Reusel	Rimburg	Sappemeer
Nuland	Onstwedde	Opsterland	Oudhuizen	Pijnacker-Nootdorp	Reusel-De Mierden	Ritthem	Sas van Gent
Numansdorp	Onwaard	Oss	Oudkarspel	Poederoijen	Rheden	Rockanje	Sassenheim
Nunhem	Ooltgensplaat	Ossendrecht	Oudorp	Polsbroek	Rhenen	Roden	Schaesberg

Nunspeet	Oost Gelre	Ossenisse	Oudshoorn	Poortugaal	Rhijnauwen	Roerdalen	Schagen
Nuth	Oost- en West-Barendrecht	Oterleek	Oud-Valkenburg	Poortvliet	Rhoon	Roermond	Schaijk
Obbicht en Papenhoven	Oost- en West-Souburg	Ottersum	Oud-Vossemeer	Portengen	Ridderkerk	Roggel	Schalkwijk
Obdam	Oost-, West- en Middelbeers	Ottoland	Oud-Vroenhoven	Posterholt	Riethoven	Roggel en Neer	Schardam
Odijk	Oost-Barendrecht	Oud en Nieuw Gastel	Oud-Wulven	Princenhage	Rietveld	Rolde	Scharsterland
Odoorn	Oostburg	Oud en Nieuw Mathenesse	Oukoop	Prinsenbeek	Rietwijkeroord	De Ronde Venen	Scharwoude
Oeffelt	Oostdongeradeel	Oud-Alblas	Ouwerkerk	Purmerend	Rijckholt	Roosendaal	Scheemda
Oegstgeest	Oosterbroek	Oud-Beijerland	Overasselt	Putte	Rijneveld	Roosendaal en Nispen	Schellingwoude
Oerle	Oosterhesselen	Ouddorp	Overbetuwe	Putten	Rijnsaterwoude	Roosteren	Schellinkhout
Ohé en Laak	Oosterhout	Oude en Nieuwe Struiten	Overschie	Puttershoek	Rijnsburg	Rosmalen	Schelluinen
Oijen en Teeffelen	Oosterland	Oude IJsselstreek	Overslag	Raalte	Rijnwaarden	Rossum	Schermer
Oirsbeek	Oostflakkee	Oude Pekela	Ovezande	Raamsdonk	Rijnwoude	Rotterdam	Schermerhorn
Oirschot	Oosthuizen	Oudelande	Pannerden	Ransdorp	De Rijp	Roxenisse	Scherpenisse
Oisterwijk	Oostkapelle	Oudenbosch	Papekop	Rauwerderhem	Rijsbergen	Rozenburg	Scherpenzeel
Oldambt	Oost-Souburg	Oudendijk	Papendrecht	Ravenstein	Rijsenburg	Rozendaal	Schiebroek
Schiedam	Sint Laurens	Sommelsdijk	Steenwijkerwold	Teteringen	Usquert	Vianen	Vrije en Lage Boekhorst
Schiermonnikoog	Sint Maarten	Son en Breugel	Stein (L.)	Texel	Utingeradeel	Vierlingsbeek	Vrijenban
Schijndel	Sint Maartensregt	Spaarndam	Stein (ZH.)	Teylingen	Utrecht (gemeente)	Vierpolders	Vrijhoeve-Capelle
Schimmert	Sint Odilienberg	Spaarnwoude	Stellendam	Tholen	Utrechtse Heuvelrug	Vijfheerenlanden	Vrijhoeven
Schin op Geul	Sint Pancras	Spanbroek	Sterkenburg	Thorn	Vaals	Vinkeveen	Vrouwenpolder
Schinnen	Sint Philipsland	Spaubeek	Stevensweert	Tiel	Valburg	Vinkeveen en Waverveen	Vught
Schinveld	Sint Pieter	Spijk	Stichtse Vecht	Tienhoven (U.)	Valkenburg (L.)	Vlaardingen	Vuren
Schipluiden	Sint-Annaland	Spijkenisse	Stiphout	Tienhoven (ZH.)	Valkenburg (ZH.)	Vlaardinger-Ambacht	de Vuursche
Schokland	Sint-Maartensdijk	Sprang	Stolwijk	Tietjerksteradeel	Valkenburg aan de Geul	Vlagtwedde	Waadhoeke
Schonauwen	Sint-Michielsgestel	Sprang-Capelle	Stompwijk	Tilburg	Valkenburg-Houthem	Vledder	Waalre
Schoondijke	Sint-Oedenrode	St. Anthonis	Stoppeldijk	Tongelre	Valkenisse	Vleuten	Waalwijk

Schoonebeek	Sittard	Stad aan 't Haringvliet	Stormpolder	Tubbergen	Valkenswaard	Vleuten-De Meern	Waarde
Schoonhoven	Sittard-Geleen	Stad Delden	Stoutenburg	Tudderen (Drostambt)	Varik	Vlieland	Waardenburg
Schoonrewoerd	Skarsterlân	Stad-Almelo	Stramproy	Tull en 't Waal	Veen	Vlierden	Waarder
Schoorl	Sleen	Stad-Doetinchem	Stratum	Twenterand	Veendam	Vliet	Waddinxveen
Schore	Slenaken	Stad-Hardenberg	Streefkerk	Twisk	Veenendaal	Vlijmen	Wadenoijen
Schoten	Sliedrecht	Stad-Ommen	Strevelshoek	Tynaarlo	Veenhuizen	Vlissingen	Wageningen
Schoterland	Slochteren	Stadskanaal	Strijen	Tytsjerksteradiel	Veere	Vlist	Wamel
Schouwen-Duiveland	Sloten (F.)	Stad-Vollenhove	Strijensas	Ubach over Worms	Veghel	Vlodrop	Wanneperveen
Schuddebeurs en Simonshaven	Sloten (NH.)	Standdaarbuiten	Strijp	Ubbergen	Veldhoven	Voerendaal	Wanroij
Serooskerke (Schouwen-Duivenland)	Sluipwijk	Staphorst	Strucht	Uden	Veldhoven en Meerveldhoven	Vogelwaarde	Wanssum
Serooskerke (Walcheren)	Sluis	Stavenisse	Súdwest-Fryslân	Udenhout	Veldhuizen	Vollenhove	Warder
Sevenum	Sluis (oud)	Staveren	Susteren	Uitgeest	Velp	Voorburg	Warffum
Sijbekarspel	Sluis-Aardenburg	Stavoren	Swalmen	Uithoorn	Velsen	Voorhout	Warmenhuizen
Simpelveld	Smallingerland	Stede Broec	Teckop	Uithuizen	Venhuizen	Voorschoten	Warmond
Sint Anna Termuiden	Smilde	Stedum	Tegelen	Uithuizermeeden	Venlo	Voorst	Warnsveld
Sint Anthonis	Sneek	Steenbergen	Tempel	Ulestraten	de Vennip	Vorden	Waspik
Sint Anthonypolder	Snelrewaard	Steenbergen en Kruisland	Terheijden	Ulrum	Venray	Vreeland	Wassenaar
Sint Geertruid	Soerendonk	Steenderen	Termunten	Urk	Verwolde	Vreeswijk	Watergraafsmeer
Sint Jansteen	Soest	Steenwijk	Terneuzen	Urmond	Vessem, Wintelre en Knegsel	Vries	Wateringen
Sint Kruis	Someren	Steenwijkerland	Terschelling	Ursem	Veur	Vriezenveen	Waterland
Waterlandkerkje	Westerveld	Wijlre	Workum	Zeewolde	Zuidhorn		
Waverveen	Westervoort	Wijnandsrade	Wormer	Zegveld	Zuidland		
Wedde	Westerwolde	Wijngaarden	Wormerland	Zegwaart	Zuidlaren		
Weerselo	Westkapelle	Wildervank	Wormerveer	Zeist	Zuidplas		
Weert	Westland	Willemstad	Woubrugge	Zelhem	Zuid-Polsbroek		
Weesp	Westmaas	Willeskop	Woudenberg	Zesgehuchten	Zuidschalkwijk		
Weesperkarspel	West-Souburg	Willige-Langerak	Woudrichem	Zevenaar	Zuid-Scharwoude		

Wehl	Weststelling werf	Wilnis	Wouw	Zevenbergen	Zuid-Waddinxveen		
Wemeldinge	Westvoorne	Wilsum	Wulverhorst	Zevender	Zuidwijk		
de Werken en Sleeuwijk	Westwoud	Wimmenum	Wûnseradiel	Zevenhoven	Zuidwolde		
Werkendam	Westzaan	Winkel	Wymbritseradeel	Zevenhuizen	Zuidzande		
Werkhoven	Wieldrecht	Winschoten	Wymbritseradiel	Zevenhuizen-Moerkapelle	Zuilen		
Wervershoof	Wierden	Winsum	Yerseke	Zierikzee	Zuilichem		
Wessem	Wieringen	Winterswijk	Zaamslag	Zijpe	Zundert		
West Betuwe	Wieringermeer	Wisch	Zaandam	Zoelen	Zutphen		
West Maas en Waal	Wieringerwaard	Wissekerke	Zaandijk	Zoetermeer	Zwaag		
West-Barendrecht	Wijchen	Wissenkerke	Zaanstad	Zoeterwoude	Zwammerdam		
Westbroek	Wijdemeren	Wittem	Zalk en Veecaten	Zonnemaire	Zwartewaal		
Westdongeradeel	Wijdenes	Woensdrecht	Zaltbommel	Zoutelande	Zwartewaterland		
Westdorpe	Wijdewormer	Woensel	't Zandt	Zouteveen	Zwartsluis		
Westerbork	Wijhe	Woerden	Zandvoort	Zuid- en Noord-Schermer	Zweeloo		
Westerhoven	De Wijk	Wognum	Zederik	Zuid-Beijerland	Zwijndrecht		
Wester-Koggenland	Wijk aan Zee en Duin	De Wolden	Zeeland	Zuidbroek (Gr.)	Zwolle		
Westerkwartier	Wijk bij Duurstede	Wolphaartsdijk	Zeelst	Zuidbroek (ZH.)	Zwollerkerspel		
Westerschouwen	Wijk en Aalburg	Wonseradeel	Zeevang	Zuiddorpe			

Table 4: Annotation Scheme for GM Label

Ambiguous words	**Surnames**	**University Names**	**Airports**	**Other**
Huizen	Van der Hoeven	Universiteit van Amsterdam	Eindhoven Airport	Adresses
Waarde	Bunschoten	Universiteit Maastricht	Luchthaven Groningen	Verdrag van Maastricht
Heel	Van den Brakel	Universiteit Utrecht		Nieuw Zeeland
Buren	Kralingen	Universitair medisch centrum Utrecht		Naturalis Leiden
	Rietveld	Universiteit Leiden		’s- Gravenhage (Den Haag)
	Van Beuningen	Rijksuniversiteit Groningen		‘s-Hertogenbosch (Den Bosch)
	Veldhuizen			

Table 5: Difficult Cases during Annotations.

Appendix IV

This README provides information on how to run the code that was used for this internship project.

README

INCREASING FINDABILITY OF LOCAL GEOGRAPHICAL INFORMATION

This repository is created for the internship project of Jasmine van Vugt at Statistics Netherlands from April 1st until June 29th.

This internship was performed for the Master Text Mining at the Vrije Universiteit Amsterdam.

GOAL

The aim of this internship project was to improve findability of local geographical information in the CerBeruS search engine.

APPROACH

To do so, first a Rule-Based Baseline was created to retrieve documents that contain local geographical data. After that manual annotation were performed and two BERT models, BERTje and RobBERT, were fine-tuned on Named Entity Recognition. Finally, those predictions in combination with the token were used to perform Named Entity Linking to link the corresponding local geographical code with the local geographical indication inside the articles.

DATA

The documents that are used for this project were the articles of the open data of Statistics Netherlands and a gazetteer which included local geographical indications. The data can be found in the data folder and include:

- Open data articles : articles.pk
- Gazetteer: Gebieden__overzicht_vanaf_1830_23032021_163423.ods

This README provides information on how to run the scripts to create a Rule-Based baseline, run the two Dutch BERT models and to perform Entity Linking.

Requirements:

Should be set first and can be found in: requirements.txt

1) Create Rule-Based Baseline

Folder: rulebasedbaseline:

1. If the requirements are set, the create_rule_based_baseline.py script is the script used to create the rule-based baseline.

2. To set the arguments for this rule_based_baseline.py script and to run the script for the rule-based baseline: rule_based_baseline.sh is created

Create Rule-Based baseline

The arguments to set for the rule_based_baseline.sh are the following:

1. --file: Articles
2. --filename: Gazetteer
3. --outputfile: the name of the output.json file with the rule-based-baseline predictions

If the arguments are set correctly, the rule_based_baseline.sh should be ran through command prompt.

An example on how to run this via command prompt is:

1. Navigate to the directory where the bash script is stored and add bash rule_based_baseline.sh:
2. For me this was:
3. D:/CBS_used_code/bash rule_based_baseline.sh

Create Json to CoNLL for Annotations

To be able to create annotations, the selected files (containing local geographical information) from the rule-based baseline were put in CoNLL format.

Arguments for create_annotation_conll.sh:

1. --filename: rulebasedbaseline.json one created with the rule_based_baseline.sh
2. --outputfile: A folder to put all CoNLL files.

If arguments set correctly:

Run create_annotation_conll.sh from command prompt

1. Navigate to the directory where the bash script is stored and add: bash create_annotation_conll.sh
2. Example: D:/CBS_used_code/bash create_annotation_conll.sh

2) Fine-tune BERTje and RobBERT

The conll data is splitted in 80% training, 10% validation and 10% test data with the:

split_folders.ipynb in the CodeforBERT folder.

Create CoNLL to JSON for Fine-tuning

In order to fine-tune BERT models, json format is necessary.

The CBSBERT folder consists of two scripts:

create_json.py and concat.json.py those scripts:

1) create json from conll (validation, training and test data)

2) concatenate multiple json files to one json file.

How to run:

1) Run from command prompt using bash script conll2json.sh

2) Redirect to the folder where the bash script is stored:

3) Example: D:/CBS_used_code/bash conll2json.sh

BERT Scripts

In the requirements it is stated how to retrieve Transformers and what further installations are needed to fine-tune BERT

Validation data:

1) validationbertje.sh and validationrobbert.sh are the two scripts to run evaluation on the validationdata

2) You can use it for experiments on number of epochs/batch size

2) Both scripts should be ran from command prompt

3) NOTE: It might take a long time to fine-tune

4) Example: D:/CBS_used_code/bash validationbert.sh

Test data:

1) testbertje.sh and testrobbert.sh are the two scripts for predictions of NER

2) Use number of epochs/batch size which performed best for the validation data

3) Both scripts should be ran from command prompt

4) Example in command prompt: D:/CBS_used_code/bash testbertje.sh

3) Entity Linking

Predictions of BERT are only the predictions but in line with articles.

In order to link the local geographical code with the correct local geographical indication the predictions of BERT should be added to the actual text again.

Concatenate Predictions and Texts

In folder Entity Linking:

1) Code for concatenating predictions of BERT and Text

2) run with bash to set arguments

3) concat_preds.sh

Perform Entity Linking

In folder Entity Linking:

1) Code for linking entities with correct local geographical code

2) run with bash to set arguments

3) entitylinking.sh

Outcome:

After performing the Entity linking articles contain predictions of BERT and the correct local geographical code for the corresponding local geographical information.

If you have any questions, you can always contact me.