Universiteit Utrecht

Faculty of Humanities

The importance of domain-specific expertise in training customized Named Entity Recognition models

Bachelor Thesis

Aniek Brandt

Artificial Intelligence

Supervisors:
Denis Paperno, Jelte Mense

July 2, 2021

Abstract

The Dutch Expertise Center of Human Trafficking and Human Smuggling aims to use online media archives to extract articles that are useful for them. One approach for this is creating a custom Named Entity Recognition model.

Named Entity Recognition (NER) is a subtask of Information Extraction (IE). Its goal is to extract certain 'named entities' from unstructured text. These entities used to only be proper names, but today NER encompasses the extraction of all important entities within a given context [17].

When creating a custom NER model, the entities that are extracted are are by definition very domain-specific. Because of this, big, annotated training corpora usually do not exist for custom NER models and annotation is done by hand. This probes the question whether annotation should be done by people with knowledge of the given domain, or by people with knowledge of NER.

In this report, a custom NER model created by using the SpaCy library is trained on a dataset that is annotated by either a fourth year AI student or employees of the Expertise Center. This was done in order to assess the importance of domain-specific knowledge in annotating data for custom NER models. Different properties of the annotated datasets are analyzed, as well as the performance of the models.

The models trained on the dataset annotated by the AI student slightly outperformed those trained on the dataset annotated by the Expertise Center, but not by a great margin. Most of all, the outcome of the research suggests a trade-off between extracting certain, extremely specific entities and creating a model that performs and generalizes well.

Keywords: Named Entity Recognition, Annotation

Contents

1 Introduction

We currently live in a world in which enormous amounts of unstructured online data are available each day. The ability to extract the useful information out of this data and using it offers great opportunities for many organizations.

The Dutch National Police does research in many AI-related fields, one of them being Natural Language Processing (NLP). The aim of the Police within this field is to be able to use online media archives to extract articles that are useful for a certain department. This can be done by using Named Entity Recognition (NER).

NER is a subtask of the more broad topic of Information Extraction (IE). In its beginning stages, IE mainly focused on extracting textual structures from unstructured data such as news articles. However, Grishman and Sundheim [6] realized that in order to obtain useful information, the semantic value of individual words needed to be taken into consideration as well, in particular the difference between different kinds of proper nouns. The process of extracting semantic information from unstructured textual data was then defined as Named Entity Recognition (NER).

Historically, NER aims to locate certain 'named entities' in unstructured text and categorize these entities. The main focus of NER was categorizing proper nouns, i.e. nouns that are written with a capital first letter, such as proper names, organization names and locations such as countries, cities or streets. Because of this, earlier research on NER only used the categories PER (person), LOC (location), ORG (organization) and MISC (other).

However, more recent research is not limited to the extraction of proper nouns. As the domain of NER continued to grow, the four entities mentioned above were no longer sufficient [17]. For some areas of expertise, other proper nouns needed to be considered, such as 'artist' or 'book title', and for other sectors extracting entities that aren't proper nouns are preferred, like 'atom' or 'DNA'. The development of models that are able to extract this type of data led to a redefinition of what NER stood for: it now encompasses the extraction of all important entities within a given context. [17]

One of the biggest challenges for this 'customized' NER is the lack of annotated text corpora to train on. For classical NER, numerous annotated datasets already exist, ranging from news articles to e-mails to scientific articles. [10] This makes supervised learning an accessible approach for creating your own NER. For customized NER models, however, this is less straightforward.

Annotating datasets by hand requires much manual work and probes the question what 'good' annotated data is. Research [13] has been done on creating rule-based 'taggers' for text corpora, but these are not easily generalized as customized NER models are by definition very domain-specific. Therefore, the best method to obtain 'good' annotated data for smaller, custom NER models remains to put in manual labor and annotating the training data by hand.

This research is interested in the annotation process of custom NER models, specifically the importance of domain-specific knowledge during the annotation process. How important is it to have knowledge on the topic of the articles in the dataset before annotating, and how important is it to have knowledge on how Named Entity Recognition works?

These questions probed the main question this research will try to answer, which is 'What is the importance of domain-specific knowledge when annotating training data for custom Named Entity Recognition models?'

This research is done in collaboration with the Dutch National Police Force and the Dutch Expertise Center for Human Trafficking and Human Smuggling. In 2020, Nijhof researched how to extract cocaine-seizure related news articles from an online news platform using a custom NER model [9]. This research led to more questions about customized NER models for other applications within the Dutch National

Police Force, in particular human trafficking.

Firstly, a literature review will be done to gain more insight in NER and its annotation process. Secondly, a custom NER model will be trained on multiple annotated news articles about human trafficking. The aim of the model will be to extract a number of predefined custom entities from the annotated articles.

The annotation will be done twice: once by officers that work at the Expertise Center, who possess a lot of domain-specific knowledge, and once by a fourth-year AI student. In addition to this, datasets from two different sources will be used. Both datasets will be about human trafficking. However, one will be provided by the Expertise Center and contains pre-selected news articles. The other dataset will be extracted from an online news platform. Both datasets will be annotated by both the officers at the Expertise Center and the AI-student, and performance will be compared.

The structure of this research is as follows. In Chapter 2, Named Entity Recognition will be explained, as well as its annotation process. Additionally, the performance measures of NER-models and the workings of SpaCy, the library used to build the NER are discussed. In Chapter 3, more information on the datasets and chosen custom entities will be provided. In Chapter 4, the implemented scripts will be explained, as well as the preparation process the data will have to undergo before it is ready for training. Chapter 5 contains the results, Chapter 6 the discussion and Chapter 7 the conclusion.

2 Named Entity Recognition

The concept Named Entity Recognition will be explained in this chapter. It will focus on the workings of the NER models, the preparation the data has to undergo before the models are ready to be trained, and lastly on how performance of NER-models is measured.

As mentioned before, Named Entity Recognition (NER) is a subtask of Natural Language Processing (NLP). Where previously only the proper nouns PER (person), LOC (location) and ORG (organization) could be extracted from text ('classical' NER), it is now possible to create NER-models that are able to extract different, domain-specific entities. These entities do not need to be proper nouns, or nouns at all, for that matter. In this thesis, such a 'custom' NER-model will be created and trained.

There are several libraries that can be used to train custom NER-models. Examples of these are AllenNLP [1], Google's BERT, or SpaCy [16]. In this thesis, the decision has been made to create the model using the SpaCy library. The main reason to use SpaCy instead of one of the other libraries is due to its accessibility and user-friendliness, while obtaining a state-of-the-art performance score [14]. The section below explains the workings of SpaCy in more detail.

2.1 SpaCy library for NER

The SpaCy library is an advanced library for NLP in Python. It has several applications, one of which is a Named Entity Recogniser. SpaCy can look at textual information on a sentence-level rather than on word-level. However, the classification happens for individual words. Therefore, the first step in every SpaCy application is tokenization. In tokenization, the input sentence is segmented into words. This is done by applying certain language-specific rules to the input sentences. An example of such a rule is that the word 'Let's' should be split up into two words: 'Let' and ''s'. Such language-specific exceptions are hard coded, kept in a list and applied as the tokenizer iterates over the sentence. After tokenization, SpaCy uses a four-step procedure to eventually predict the entities in unseen text.

The first step after tokenization is the *embedding* of the tokenized words into binary vectors. This means that every word is assigned an ID containing only 0's and 1's. This word-specific ID is created by combining the binary representations of four of the word's characteristics into one single ID. The four characteristics that are taken into account are the norm of the word (the word itself but all letters are de-capitalized), the prefix and suffix of the word (the first three and last three letters) and the shape of the word, meaning all letters are converted to a w (or a W if it is capitalized) and all digits to a d. All these four characteristics are represented in a binary manner based on ASCII characters, and then combined into one single ID. Because of this, the chance of two words ending up with the same vector is close to zero.

After embedding the words into vectors, vectors from the same sentence are *encoded* together in a sentence matrix. In this matrix, a certain token and its neighboring tokens (that often contain semantically useful information) are grouped together. This is done by feeding the embedded ID's into a neural network that combines every token ID with the token ID of its two direct neighbors, and then the token ID of the neighbors next to the direct neighbors, and so forth. The SpaCy encoding neural network is four layers deep, meaning a word and its 4 closest neighbors on each side of the word are taken into consideration. After the encoding step, a multi-dimensional matrix is left as the context-including representation of the sentence.

In the third step, *attend*, the multi-dimensional matrix of a certain token is

shrunk back to a one-dimensional vector. However, contrary to the vectors after the embedding step, the vector of a token now contains context-sensitive information. Finally, using this vector, the model assigns a classification to the words using a feed-forward neural network and the entity is *predicted*.

SpaCy is a statistical library, meaning predictions are based on the *probability* of a word being a certain entity. These probabilities are tweaked in the training process. To illustrate, during the training process the model should learn that a capitalized word following the word 'from' is most likely a location of origin and not a financial quantity. All steps of the classification process are shown in Figure 1 below:

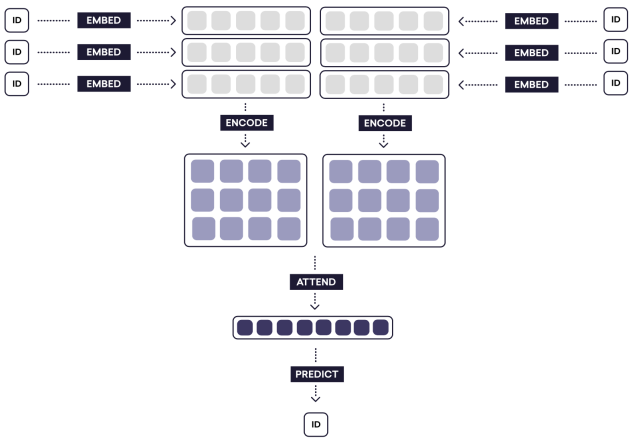


Figure 1: The steps of classification used in a SpaCy NER.

2.2 Performance Measures

After annotation, the NER models are ready to be trained and tested. Training the models will be done by using k-fold cross validation. *K-fold cross-validation* is a technique often used for the training of Machine Learning (ML) models in which the original training data is partitioned into k equal sized subsets [12]. One of these samples is kept as a test set, while the other k-1 samples are used to train the model. The entire training process is repeated k times, so each of the subsets is used as the validation set exactly once. The most commonly used numbers for k are 5 and 10 . In between the training of different k-folds, the model can change and tweak its parameters. After this, the best performing model is selected and used on the test set. The performance of the test set on the model can be used as a general performance measure of the model.

To evaluate the performance of the test set on the models, the precision, recall and F1-score will be used. These are the most commonly used evaluation metrics in text classification [7]. *Precision* measures the correctly classified entities amongst all detected entities. It is defined as:

$$Precision = \frac{true\ positives}{true\ positives\ +\ false\ positives}$$

Recall measures the percentage of entities that were detected by the model that were actually annotated as such by humans. It is defined as:

$$Recall = \frac{true\ positives}{true\ positives\ +\ false\ negatives}$$

In most text classification models, the precision and recall of the model are equally important. Therefore, it is useful to combine the two into a single metric. There are several ways to do this, but the simplest form is the *F1-score* [11]. The F1-score essentially is the harmonic mean of the precision and recall, and it is defined as:

$$F1 = 2 * \frac{precision\ *\ recall}{precision\ +\ recall}$$

In this research, the F1 score will be used as the main measure of performance.

3 Data

In this chapter, the text corpora on which the model will be trained will be further explained as well as the custom entities the model will be trained on. All of the decisions regarding the data sources have been made in collaboration with the Dutch Expertise Center for Human Trafficking and Human Smuggling. The aim of this research is to provide insight in how they can use NER to gain a broader insight in the developments in human trafficking on a global scale. This is why the data sources and custom entities are tailored to their domain and its applications.

3.1 Data sources

The two main data sources used in this research are the Europe Media Monitor and the so-called MediaScans, that consist of a collection of news articles already used by the Expertise Center. These sources will be used to set up two text corpora that the custom NER will use as test- and train data. In this subsection, both data sources and their specifications will be explained further.

3.1.1 MediaScans

To stay up to date on the developments of the human trafficking field, the Expertise Center uses MediaScans, a collection of relevant articles from different newspapers, both English and Dutch. These articles are scraped from LexisNexis, an online database that collects data from over 200 different sources, both Dutch and international [4]. Using search queries that match on certain keywords, a selection of relevant articles is made and collected into the MediaScans. These queries are set up for each relevant sector, meaning there are separate queries for every aspect of human trafficking. The queries total a number of seven sectors, including 'forced prostitution' and 'forced organ donation'.

The Expertise Center already uses the data from the MediaScans themselves, making it a relevant source of data for this research. It is valuable to know how the custom NER-model performs on articles of which we already know they are domain-relevant, as this allows the model to be trained on data of guaranteed quality.

3.1.2 Europe Media Monitor

Whilst the queries from LexisNexis are a valuable source of data for the Expertise Center, they would like to broaden the scope of the data. The MediaScans only contain news articles, yet other sources such as blogs could also be useful. Additionally, the main focus of the MediaScans are Dutch news articles while a more international perspective could also be of value. This is where the Europe Media Monitor comes into the picture.

The Europe Media Monitor is an incentive of the European Commission's Joint Research Centre. It extracts information from over a thousand news sources including newspapers, blogs and websites, in over 70 languages. [3] The information extracted from these sources is categorized by topic. One of these topics is human trafficking, thus making the Europe Media Monitor interesting to use for the Expertise Center.

However, as the queries for the MediaScans run continuously, enormous amounts of data are collected each day. Meetings with employees of the Expertise Center brought forward that reviewing the queries and selecting the most relevant articles is something that should be done at least every few days, in order to prevent that the number of articles gets too high to process. Using queries on the Europe Media Monitor and adding those articles to the already existing MediaScans would create

a workload that is no longer manageable. It is possible for the Expertise Center to use either the MediaScans or the Europe Media Monitor, but not both at once.

Therefore, it would be useful to know which is the better, more complete source. For this, it is useful to use Named Entity Recognition instead of keyword-based queries, as NER is sentence-based instead of word-based. This makes it possible to get a better understanding of the difference between the two sources on an objective, statistical level. This is put in contrast with using queries and comparing the articles extracted manually, which is more subjective and error-prone.

3.2 Custom entities

In order to be able to train the custom Named Entity Recognition model, custom entities need to be selected. This has been done in in consultation with the Expertise Center as well, over a series of meetings. The following twelve entities were selected:

Entity	**Examples**
Victim	underage girls, babies
Suspect	pimp, trafficker
Sector	prostitution, sexual exploitation
Location of origin	Eritrea, Albania
Location of destination	London, Amsterdam
Documents	passport, visa
Transport	train, boats
Circumstances	taking explicit photos, work without break
Organization	police, Children's Aid
Financial quantities	500.000, thousands
Financial units	$, dollars

Table 1: Selected custom entities.

In selecting these entities, the difference between domain-specific expertise and NER-related expertise already became apparent. NER is a text-driven extraction method, making it difficult to extract lengthy entities or entities that do not occur more than once within the same context. Even so, information from exactly those entities prove to be useful to the Expertise Center. This is why not all are expected to perform equally well.

For some entities, such as 'suspect' and 'victim', the model is prone to overfitting, especially considering the limited size of the dataset. Examples of overfitting could be always categorizing *women* as 'victim' and *men* as 'suspect', while this is not always the case.

The entity 'circumstances' is difficult as well, as it is a very broad entity. The same instance of a circumstance is unlikely to occur more than once, which can make it hard for the model to generalize. However, this entity in particular is useful for the Expertise Center as conditions upon finding are very important in categorizing whether something is an instance of human trafficking or not.

Other entities, such as 'documents' and 'transportation', are not expected to occur much at all. However, extracting them when they do occur is important for the Expertise Center, which is why the model is to be trained on them.

3.3 Annotators

In coordination with the Expertise Center, the decision was made to annotate 20 articles from both sources, totaling 40 articles for training and validation. In addi-

tion to this, 4 articles from both sources were annotated to use as a test set. The in total 48 articles were annotated by employees of the Expertise Center as well as a fourth year AI student, leading to a total of 4 annotated datasets of 24 articles each.

The annotation from the Expertise Center was done by two employees. They did not work together but instead divided the work, meaning no article was covered twice. Both employees have a long history of working in human-trafficking related organizations and thus possess a lot of domain-specific knowledge.

The student that annotated the articles without having any domain-specific knowledge has followed numerous university-level courses that covered the subject of NLP, and thus has an excellent grasp of how annotation works in the technical sense.

4 Methods

The following section will provide a description of the different stages of research that need to be carried out to answer the research question. Firstly, data collection and the transformations performed on the data will be explained. Afterwards, a further description of the models created with the data will be provided.

4.1 Data preparation

4.1.1 Data collection

In this subsection, the data collection process will be explained. As explained in section 3.1, this research used two different sources of datasets. The dataset from both sources were annotated by two groups of annotators, resulting in a total of four annotated datasets. Both datasets consist of 24 news articles each. This number was chosen due to the limited timeframe of this research.

The first dataset consists of 24 articles selected from the MediaScans (MS). The selection was done by the employees of the Dutch Expertise Center for Human Trafficking and Human Smuggling. 16 of these articles were in Dutch, where the other 8 articles were written in English. The SpaCy NER performs best on English articles, which is why the Dutch articles were translated to English using the DeepL translator. [5]

A python script was then run in order to split each article by sentence, which made them easier to annotate. After this, the MS-dataset was sent back to the Expertise Center and the annotation process was started. The AI student annotated the MS-dataset as well. This resulted in two annotated MS-datasets: **MS_AI** and **MS_POL**

The second dataset is a set of news articles collected from the Europe Media Monitor (EMM). For the web scraping of these articles the python scripts from Nijhof [9] were used as a baseline. Articles that were scraped were English articles, published between May 1st, 2021 and May 5th, 2021 that contained the words 'human traffic'. Articles that contained the words 'migration' and 'smuggling' were excluded, as this research focuses on human traffic only. In total, over a hundred articles were scraped. From these articles, a selection of 24 articles was made by the Expertise Center. The selected articles were then split by sentence, sent back to the Expertise Center and annotated the same way as the MS-dataset. The EMM-dataset was annotated by the AI student as well. The two annotated EMM-datasets are referred to as **EMM_AI** and **EMM_POL**.

The annotators from the Expertise Center indicated that the preparation of the annotation process (including meetings, deciding on the entities that were to be extracted, etc.) took them two days. After this, the annotation of the articles took roughly 45 minutes per article. For a total of 48 articles (24 from each data source), this equaled 36 hours of work.

This is slightly more than the AI-student, who annotated one article roughly every 30 minutes, making their annotation process 24 hours long.

4.1.2 Data transformation

After the annotation process, a transformation of the datasets was needed. Both the MS- and EMM-dataset were annotated by using SGML-tags with the custom entities as start- and endtags, and the selected text as the body. This method of annotating was chosen because it is very intuitive, which sped up the annotation process. One such annotated sentence is shown below:

<org>Police</org>want a ban on the anonymous purchase of <comm>pay-as-you-go mobile phones</comm>to help tackle county lines <person_suspect>drug dealers</person_suspect>, a report reveals.

While annotating using SGML-tags is very user-friendly, SpaCy requires its train data to be in a different format. A python script was written and run on the four annotated datasets to acquire four lists of train data (one for every dataset), ready to be used by SpaCy. The annotated sentence below shows the same sentence as above, but in the correct input-format for the SpaCy model:

('Police want a ban on the anonymous purchase of pay-as-you-go mobile phones to help tackle county lines drug dealers, a report reveals.', {'entities': [(0, 6, 'org'), (47, 74, 'comm'), (103, 115, 'person_suspect')]})

After transformation of the data the SpaCy models were ready to be trained and tested.

4.2 Analysis

4.2.1 Data analysis

Because the focus of this thesis is the annotation process in creating a custom NER-model , the importance of extensive analysis of the annotated data (before it is even processed by the model) cannot be overlooked. In this subsection the conducted tests on the annotated data will be explained.

The properties of the annotated data that have been analyzed are entity length, entity frequency and entity density, the latter of which is defined as the number of entities per article.

For the analysis of the entity length, three two-factor ANOVA tests have been conducted in SPSS. The tests were done to see whether the annotations by either annotator (POL or AI) were significantly longer than those by the other annotator. Difference in annotation length can be a good predictor for the performance of the NER-models, as NER is sentence-based and thus takes the context of a word into consideration when assigning an entity to it. In the first two ANOVA tests, the annotator (POL or AI) and the twelve different entities were used as (categorical) independent variables, and the entity length was be used as the (continuous) dependent variable. This test was conducted twice as it was done for both the EMM dataset and the MS dataset individually.

In the third test, the source of the dataset (EMM or MS) and the twelve different entities were used as independent variables, and entity length was be used as the dependent variable. While the annotator was not considered when comparing the data sources, this test helped gain insight in the general qualities and differences of the two data sources.

The entity frequency was analyzed by counting the occurrence of each of the twelve entities in each of the four datasets and comparing them.

To compute the density per article, the number of entities in each of the twenty articles in the four datasets were collected and plotted. Comparison was done not only between data sources, but between annotators as well.

4.2.2 Model analysis

As explained in subsection 4.1, the data preparation process resulted in four lists of annotated data, in the correct format, ready to be used as training sets by SpaCy. In this subsection, the training, testing and validation process will be explained in detail.

After collection and transformation, the data was ready to be used to train the SpaCy NER models. The models were trained for each of the four datasets individually. Training was done using a 5-fold cross-validation on 20 of the articles, where 4 articles were kept aside as a test set.

Out of the five versions of the model the k-fold generated, the model with the highest F1-score was kept as the model to be used on the test sets. This means that after training and validation, four models in total were kept for testing:

- **EMM_POL**, the model generated from the Europe Media Monitor data annotated by the Expertise Center.
- **EMM_AI**, the model generated from the Europe Media Monitor data annotated by the AI student.
- **MS_POL**, the model generated from the MediaScans data annotated by the Expertise Center.
- **MS_AI**, the model generated from the MediaScans data annotated by the AI student.

While all models were trained and validated on their own annotated data, in order to see what the importance of domain-specific expertise is, testing is done cross-annotator as well. When comparing different annotations this is especially important. Which models will be used on the different test sets are explained in Table 2.

Model	**Test set**
EMM_AI	EMM_AI
EMM_AI	EMM_POL
EMM_POL	EMM_POL
EMM_POL	EMM_AI
MS_AI	MS_AI
MS_AI	MS_POL
MS_POL	MS_POL
MS_POL	MS_AI

Table 2: Overview of models and test sets used in the research.

For every combination of model and test set, both the overall F1 score and F1 score per entity will be calculated. This will give an indication of the performance of the model.

As seen in Table 2, every model is used on its own annotated data, as well as on the annotated data from the same source but a different annotator. This way, the generalization level of both the model and the annotation can be computed; if a model performs equally well on its own annotated data as the data annotated by the other annotator, it can be an indicator that the model generalizes well. It is desirable for a model to be able to generalize well, because this indicates that the model is not very biased and the annotations are more broadly applicable. The performance of the models on their own test set can be used as a baseline of performance.

5 Results

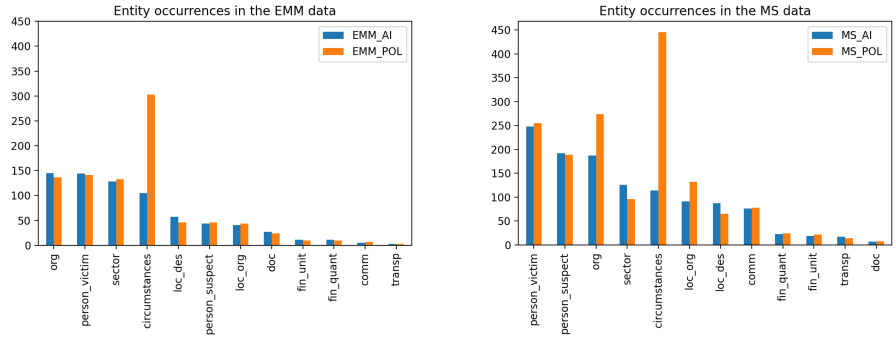
In this chapter, the results of the research will be presented. This chapter will be divided into two sub-chapters: firstly, the results of the data-analysis are covered. Secondly, the performance of the models on the test sets are evaluated.

5.1 Data-analysis results

For the analysis of the annotation data, only the 20 articles used to train the data with k-fold cross validation were used. The test set was kept separate. The reason for this is that the model is not trained on the test set; the following section describes the properties the model was trained on.

5.1.1 Entity frequency

For comparing the entity frequencies, a python script was written and executed that summed up all occurrences of every entity within a dataset. The following plots were generated (see Figure 2):



(a) EMM dataset **(b) MS dataset**

Figure 2: The occurrence of every entity by annotator by dataset. *The twelve different entities are displayed on the x-axes and the entity count is displayed on the y-axis.*

In the figures it can be seen that while the frequency of some entities differ greatly between the two different datasets, the frequency within datasets is often quite similar. This indicates that there were no big disagreements in identifying entities between the AI student and the annotators from the Expertise Center. One notable difference is the 'circumstances' entity. In this case, the entity is used by the Expertise Center more than three times than the occurrence level of the AI student in the data from both sources. This can cause a lot of noise when training and evaluating the models trained on those datasets. In addition to this, it can also be seen that the EMM dataset contains less instances of every entity than the MS dataset, which can influence the performance of the models as well.

5.1.2 Entity density

The third property of the annotated data that was analyzed in this research was the entity density. We define entity density as the number of entities per article. This analysis is not particularly useful to illustrate the difference between the annotators, but rather the difference of usefulness of different articles from different data sources.

Figure 3 was generated, where the x-axis shows the different annotated datasets, and the y-axis the number of entities in each article.

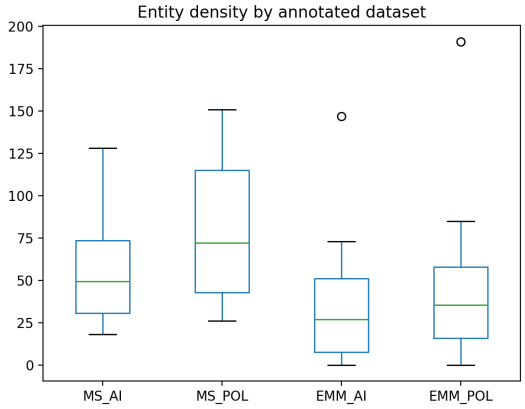


Figure 3: Entity density by dataset. *On the x-axis the different datasets are displayed. On the y-axis the amount of annotated entities per article in that dataset are shown.*

From this boxplot it becomes clear that the entity density differs greatly between different datasets. It is shown that whilst MS articles almost always contain at least around 30 entities, the average number of entities in the EMM dataset is much lower. Some articles that were particularly high in entity density are considered outliers. A reasonable explanation for this can be found in the different ways the EMM and MS datasets were obtained. The MS articles were pre-selected, whereas web scraping was used to extract the EMM articles. This is more error-prone and more non-relevant articles (or even incomplete articles) could have gotten into the dataset, which might have led to little to no relevant entities in some articles.

Aside from entity density, the eventual absolute size of the training sets was evaluated as well. Whilst the inherent datasets were all equally sized (20 articles), different article lengths and different annotators caused the eventual training sets to be different in size. The size of the datasets are defined as both the number of individual entities in the training set (n), as well as the total number of sentences that contain one or more entities (N). The total number of sentences is relevant because the SpaCy NER is able to extract entities from the context of a sentence, which is why data is separated by sentence as well. The size of the training sets of each dataset is displayed in Table 3: The numbers in this table are important to

		n	N
EMM	POL	904	324
	AI	722	284
MS	POL	1604	587
	AI	1188	495

Table 3: Size of the train- and test set for all models

consider when interpreting the performance of the models, as the amount of data each model trained on differs greatly, not only between datasets but also between annotators. In the EMM and MS dataset, the sets annotated by the AI student respectively only contained 79.8% and 74.0% of the number of entities the POL datasets contained. This can influence the performance of the models trained on

these datasets. However, the critical note should be made that the 'circumstances' entity makes up a considerable percentage of this difference. Additionally, Table 3 shows that the EMM training set is smaller than the MS dataset, which could lead to a difference in performance as well.

5.1.3 Entity length

In order to compare the entity length between annotators and between data sources, three two-way ANOVA tests have been conducted. In the first and second test, the annotator and different entities have been used as independent variables, with the length as the dependent variable. The test was done twice, as the two data sources (EMM and MS) were kept separate. The descriptive statistics of those two tests are shown in table 4 and 5:

	circumstances		comm		doc		fin_quant		fin_unit		loc_des	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
AI	16,94	11,62	8,20	5,14	11,21	5,04	5,75	3,14	1,00	0,00	8,36	3,20
POL	19,52	14,71	13,08	9,69	11,92	5,27	4,45	1,97	1,00	0,00	8,70	3,56

	loc_org		org		person_suspect		person_victim		sector		transp	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
AI	8,07	2,95	14,21	13,76	10,88	5,73	9,34	5,85	13,94	4,78	9,50	2,89
POL	8,70	3,54	16,29	11,45	12,16	8,03	9,54	7,67	14,07	5,88	10,33	2,89

Table 4: Descriptive statistics of two-way ANOVA on EMM data. *The two-way ANOVA on the EMM data showed no significant interaction effect between the length of the different entities and their annotators ($F(10) = 0.561$, $p = 0.861$). However, the entities did differ significantly in length between themselves ($F(11) = 23.704$, $p < 0.005$). This is in line with the expectations, as circumstances are usually way longer than financial quantities, for example. There was no significant difference found between the annotator and the entity length ($F(1) = 1.304$, $p = 0.254$).*

	circumstances		comm		doc		fin_quant		fin_unit		loc_des	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
AI	18,55	10,29	9,92	4,63	13,00	9,785	7,03	3,13	3,29	2,01	8,40	3,70
POL	25,53	17,33	12,40	8,04	27,75	11,87	5,80	2,45	3,61	2,62	9,59	4,94

	loc_org		org		person_suspect		person_victim		sector		transp	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
AI	7,68	2,94	13,43	10,61	9,36	4,67	9,47	5,39	14,01	3,99	6,65	3,10
POL	8,24	3,37	13,53	11,09	10,05	6,05	9,80	5,99	14,90	7,71	11,79	6,65

Table 5: Descriptive statistics of two-way ANOVA on MS data. *In the MS data an interaction effect was found between the different entities and the annotators ($F(10) = 5.703$, $p < 0.001$). In addition to this, not only did the entities individually differ significantly in length ($F(11) = 61.724$, $p < 0.001$), but the annotations done by the different annotators differed significantly as well ($F(1) = 18.017$, $p < 0.001$).*

In Table 4, it is shown that there was no significant difference found between the annotator and the entity length. This means that the annotator (POL or AI) cannot be well-predicted by the entity length. In contrary, in the MS data shown in Table 5, the annotations done by different annotators did differ significantly in length. This means that for the MS data, the annotations done by the Expertise Center (POL) were significantly longer than those done by the AI student (AI).

A third two-way ANOVA test was done to check for a possible difference in length between the two data sources. In conducting this test, no distinction was

made between the different annotators. The different entities and data sources were used as independent variables, and entity length was used as the dependent variable. The descriptive statistics are shown in Table 6:

	circumstances		comm		doc		fin_quant		fin_unit		loc_des	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
EMM	18,85	14,01	10,86	8,17	11,53	5,11	5,13	2,67	1,22	1,04	8,50	3,34
MS	23,97	16,28	11,18	6,69	19,94	12,93	6,46	2,87	3,45	2,31	8,92	4,32

	loc_org		org		person_suspect		person_victim		sector		transp	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
EMM	8,39	3,26	15,23	12,71	11,43	6,82	9,44	6,79	14,01	5,33	9,86	2,67
MS	8,00	3,19	13,49	10,88	9,70	5,67	9,63	5,69	14,41	5,96	13,38	11,21

Table 6: Descriptive statistics of two-way ANOVA on both EMM and MS data *The ANOVA test shows that there is a significant interaction effect between data source and entity and the length of the entity length ($F(10) = 7.970$, $p < 0.001$). Both the individual entities as well as the different data sources showed a significant difference with respect to the entity length ($F(11) = 97.144$, $p < 0.001$ and $F(1) = 4.109$, $p < 0.05$ respectively).*

The table shows a significant interaction effect between the entity length and the two different data sources. This means that the source of the dataset is a good predictor for the entity length; annotations in the MS dataset are significantly longer than annotations in the EMM dataset.

The implications of the three ANOVA tests results for the performance of the models could be the following: because the length of the annotations done by either AI or POL only differs significantly in the MS dataset, this can be interpreted as in an indication that the models trained on the EMM dataset will perform better when tested on the model of the other annotator, because the two datasets are more alike. In short, the EMM models are expected to generalize better than the MS models.

5.2 Model results

In this section, the performance of the custom NER models will be evaluated. Because testing was only done cross-annotator but not cross-dataset, the results of the EMM models and the MS models will be discussed separately.

5.2.1 EMM models

As explained in previous chapters, a total set of 4 articles was used as a test set to measure the performance of the models. The two EMM models, EMM_POL and EMM_AI, were both tested on both test sets EMM_POL and EMM_AI, resulting in four evaluations in total. For evaluation, the precision, recall and F1 score were calculated both by entity and in total. The results are presented in the tables 7 to 10 on the following page. The rightmost column (N) represents the number of entities in the test set. Not all entity scores are present in the tables, as not all entities appeared in the test set, or, if they appeared, were not always correctly recognized by the model at least once. In calculation of either the precision or recall these events would lead to a division by zero, which is why these entities are not considered in the performance measures of the model.

Entity	Precision	Recall	F1	N
org	0,2727	0,2143	0,24	14
comm				5
person_suspect.				4
person_victim	0,5667	0,6071	0,5862	28
sector	0,5745	0,6585	0,6136	41
loc_org	0,75	0,2308	0,353	13
loc_des				0
transp				0
doc				0
fin_quant				1
fin_unit				1
circumstances	0,129	0,0556	0,0777	73
Total	**0,429**	**0,302**	**0,354**	**180**

Table 7: Performance measures of the EMM_POL model on the EMM_POL test set

Entity	Precision	Recall	F1	N
org	0,3636	0,3077	0,3333	13
comm				5
person_suspect.				22
person_victim	0,5862	0,5	0,5397	34
sector	0,7609	0,5833	0,6604	60
loc_org	0,75	0,2143	0,3334	14
loc_des				9
transp				1
doc				2
fin_quant				1
fin_unit				1
circumstances	0,0968	0,1154	0,1053	27
Total	**0,5**	**0,328**	**0,396**	**190**

Table 8: Performance measures of the EMM_POL model on the EMM_AI test set

Entity	Precision	Recall	F1	N
org	0,2083	0,3846	0,2702	13
comm				5
person_suspect				22
person_victim	0,6364	0,6176	0,6269	34
sector	0,8462	0,7333	0,7857	60
loc_org	0,6	0,6429	0,6207	14
loc_des	0,4	0,2222	0,2857	9
transp				1
doc				2
fin_quant				1
fin_unit				1
circumstances				27
Total	**0,604**	**0,429**	**0,502**	**190**

Table 9: Performance measures of the EMM_AI model on the EMM_AI test set

Entity	Precision	Recall	F1	N
org	0,0909	0,1429	0,1111	14
comm				5
person_suspect				4
person_victim	0,5758	0,6786	0,623	28
sector	0,6538	0,8293	0,7312	41
loc_org	0,4667	0,5385	0,5	13
loc_des				0
transp				0
doc				0
fin_quant				1
fin_unit				1
circumstances				73
Total	**0,466**	**0,346**	**0,397**	**180**

Table 10: Performance measures of the EMM_AI model on the EMM_POL test set

From the tables it can be seen that all models perform similarly. The overall performance of the models is not very high, as it is compromised by certain individual entities that perform poorly. The AI model, used on its own test set, is the best performing combination, but its difference with the other three combinations is small. When considering the performance of the individual entities, a maximum of five entities (out of twelve) were extracted in total.

5.2.2 MS models

Equivalent to the EMM models, the two MS models, MS_POL and MS_AI, were evaluated on both their own test set and the test set annotated by the other annotator. Precision, recall and F1 score were calculated and are shown in Table 11 to 14 on the following page, together with the number of entities in the test set (N):

Entity	Precision	Recall	F1	N
org	0,3846	0,3125	0,3448	32
comm				0
person_suspect	0,5769	0,6	0,5882	25
person_victim	0,5714	0,4848	0,5245	33
sector	0,6875	0,5238	0,5946	21
loc_org	0,5	0,7917	0,6129	24
loc_des	0,3333	0,0667	0,1112	15
transp				0
doc				0
fin_quant	0,5	1	0,6667	1
fin_unit	0,3333	1	0,5	1
circumstances	0,2439	0,1515	0,1869	67
Total	**0,459**	**0,385**	**0,419**	**218**

Table 11: Performance measures of the MS_POL model on the MS_POL test set

Entity	Precision	Recall	F1	N
org	0,2593	0,1707	0,2059	41
comm				0
person_suspect	0,48	0,5217	0,5	23
person_victim	0,6552	0,3725	0,475	51
sector	0,625	0,5556	0,5883	18
loc_org	0,5897	0,7188	0,6479	32
loc_des	0,3333	0,0625	0,1053	16
transp				0
doc				2
fin_quant	0,8	0,6667	0,7273	6
fin_unit	1	1	1	5
circumstances	0,15	0,1765	0,1622	35
Total	**0,460**	**0,379**	**0,416**	**229**

Table 12: Performance measures of the MS_POL model on the MS_AI test set

Entity	Precision	Recall	F1	N
org	0,3077	0,1951	0,2388	41
comm				0
person_suspect	0,7273	0,6957	0,7111	23
person_victim	0,675	0,5294	0,5934	51
sector	0,3043	0,3889	0,3414	18
loc_org	0,84	0,6562	0,7368	32
loc_des	0,4286	0,1875	0,2609	16
transp				0
doc				2
fin_quant	1	0,3333	0,5	6
fin_unit	1	0,4	0,5714	5
circumstances	0,3333	0,1176	0,1739	35
Total	**0,536**	**0,393**	**0,453**	**229**

Table 13: Performance measures of the MS_AI model on the MS_AI test set

Entity	Precision	Recall	F1	N
org	0,2174	0,1562	0,1818	32
comm				0
person_suspect	0,6364	0,56	0,5958	25
person_victim	0,425	0,5152	0,4658	33
sector	0,4091	0,4286	0,4186	21
loc_org	0,6957	0,6667	0,6809	24
loc_des	0,7143	0,3333	0,4545	15
transp				0
doc				0
fin_quant	1	1	1	1
fin_unit	1	1	1	1
circumstances	0,3333	0,0606	0,1026	67
Total	**0,458**	**0,330**	**0,384**	**218**

Table 14: Performance measures of the MS_AI model on the MS_POL test set

From the tables it is shown that the models all perform similarly, and when comparing performance between the EMM and MS models, performance is also quite similar. It is shown that the AI model tested on the AI test set has the highest F1 score out of the four combinations. Unlike the EMM models, however, the POL model on the POL test set performs the second best, but again, differences are small. When considering the performance of the individual entities, more entities were extracted than in the EMM dataset, but between annotators, the same number of entities was extracted.

6 Discussion

The aim of this research was to investigate the importance of domain-specific knowledge when annotating training sets for custom Named Entity Recognition models, applied to human trafficking. This was done by manually annotating two datasets from different sources of 24 news articles each, and training a custom SpaCy NER model on 20 of the annotated articles. Annotation was done by both a fourth year AI student without knowledge on human trafficking, and employees of the Dutch Expertise Center for Human Trafficking and Human Smuggling, who possess a lot of domain-specific knowledge. In total, four models were created, two for each annotator and each data source. The models were then evaluated on both a valuation set that was annotated by the same annotator that annotated the training data and the other annotator. Performance measures were then compared. In valuation, the models from the two different data sources were kept separate.

In evaluating the frequency of every entity in every dataset, it became clear that the 'circumstances' entity occurred three times as much in the POL data as it did in the AI data. Because this entity was the longest entity as well, this was expected to cause a lot of noise when training the POL models, which could compromise their performance.

The most important takeaway from the analysis of the entity density and entity count was the great variance in number of relevant entities between articles. This is relevant, because especially in small datasets, one 'empty' article can lead to a big dent in training size.

Firstly, the results of the data analysis are discussed. Four properties of the annotated datasets were evaluated: length, frequency, density and count. After evaluating the results of the data analysis, a better expectation of the performance of the models could be made. Based on the MS models containing significantly longer entities than the EMM models, the cross-evaluated instances of the EMM models were expected to perform better than the cross-evaluated instances of the MS models, as the training data of the AI and the POL dataset were more alike in the EMM model.

When evaluating the performance of the models, it became clear that all models, from both datasets and both annotators, performed similarly. On both datasets, the models that were trained on the data annotated by the AI student outperformed those trained by the Police, but not by a great margin. Datasets that were tested on the same dataset they were trained on also performed slightly better than those that were cross-evaluated. Overall, the overall performance of the models was quite low, and for the EMM dataset only a maximum of five out of twelve entities got recognized by the model. There are several explanations for the performance of the models.

Firstly, the size of the test set (and thus also the training set) should be considered. Some entities, for example 'transp' and 'doc', never appeared more than once or twice in the test set. As can be seen in 5.1.2, they do not appear more than fifty times in the training data either. Little training data compromises the performance of the model on that certain entity, causing overall performance to drop as well. When training a model on a larger dataset, problems such as entities being underrepresented in the test set should no longer be a factor, and a more fair evaluation of the model's performance can be done.

Secondly, for some entities, in specific 'person_suspect' 'person_victim' and 'loc_org' 'loc_des', poor performance can be attributed to the entities being too similar: the model is not able to differentiate between them. This is most apparent in the MS dataset annotated by the AI student: 22 'person_suspects' were annotated, but none were classified correctly; they were all classified as 'person_victims'. Wrongly classified entity duos like this compromise overall performance as well, as

the suspects being recognized as victims brings down not only the performance of the 'person_suspect' entity itself, but that of the 'person_victim' entity too. This is a problem that could have been avoided beforehand, if other entities had been chosen. We will further elaborate on this further on in the discussion.

A third reason for the overall low performance of the model is the 'circumstances' entity in the POL models. As already stated in paragraph 5.1.2, this entity appeared more than three times as much in the POL datasets than it did in the AI datasets. This, in combination with its length and context-reliability, makes it a difficult entity to extract. The performance measures of both datasets show that, when this 'circumstances' entity is extracted at all, the performance is much lower than the overall performance of the model, even though it has the most annotations out of all entities. This could implicate that similar to the issue where some entities are too much alike, other entities might be better to avoid altogether. Such entities are those that are long and rely heavily on the context of the situation, which makes them hard for the model to generalize. This should be considered in the entity decision process as well.

While suggesting that some entities should be defined differently or not at all could improve the overall performance of future models, it remains the question whether this is desirable. In the case of human smuggling, the distinction between victims and suspects is of great importance, as are the circumstances the victim had to endure. One suggestion for this specific case could be extracting the 'duos' as one entity, and later differentiating between them by use of a different classifier. Differentiating better between similar entities in a custom NER could be an interesting approach for further research in this area.

Because all defined entities are important in the domain of expertise, stating that some entities are better to avoid at all might be a bit short-sighted. In fact, it is precisely these entities where a difference can be seen between annotators that have knowledge of NER and annotators that have knowledge of the domain of expertise.

In this specific research, the 'circumstances' entity performed very poorly. This was the case in both the POL and AI models. However, it can be stated that while the AI models were trained on about 33% of the instances of the 'circumstances' entity, performance measures of this entity were quite similar to those of the POL models. This could be an indication that in this particular case, the annotation style and knowledge of NER of the AI student could have been an advantage, and performance might improve if the models were to be trained on a bigger dataset.

From the above results and its implications, it becomes clear that the choice of entities is an extremely important factor in defining the performance of a custom NER. Because the domain of expertise is heavily context-based and NER is essentially sentence-based, the process of deciding which entities to use proved to be a challenge. In choosing what entities to extract, it is important to consider the context in which the entities are used, and to be wary of deciding on entities that induce bias.

One example of this is the 'circumstances' entity mentioned above, but for example the 'fin_quant' proved to be difficult as well. All financial quantities are numbers, and only numbers that are financial quantities are relevant for the Expertise Center, but not all numbers are financial quantities. This can lead to apparent 'inconsistencies' in the training data, as some numbers are annotated and others are not. The SpaCy model, however, is unable to differentiate between these different numbers.

Another example of such a difficult entity are the 'person_victim' and 'person_suspect' entities. In the articles included in the datasets, men were often annotated as suspects and women were annotated as victims. While gender bias is a well-known problem in Machine Learning [2], there is little to no literature on its effect on NER. Therefore, an interesting approach for further research could be

implementing one of the mitigations proposed by Sun, Gaut en Tang [15] into a custom NER model.

One of their approaches to debias the performance of NLP models is to debias the datasets the models are trained on. This can be done by creating an augmented dataset, meaning every female pronoun is swapped for a male pronoun and vice versa. Additionally, all names that are used in the text are replaced by anonymized entities, such as 'E1'. In an augmented dataset, the sentence 'Mary likes her friend Carol' would look like 'E1 likes his friend E2'. In [8], Lee showed that using augmented training datasets, the difference in F1 scores on pro-stereotypical test sets and anti-stereotypical test sets were lowered significantly. This indicates that the model itself is less gender-biased.

Creating an augmented dataset and anonymizing the names used in the text could be an interesting addition to the custom NER models created in this research.

7 Conclusion

In this thesis, datasets of news articles on human trafficking were annotated by both domain experts from the Expertise Center for Human Trafficking and Human Smuggling and a fourth year AI student. With these annotated datasets custom NER models were trained and their performance was compared in order to answer the following research question: 'What is the importance of domain-specific knowledge when annotating training data for custom Named Entity Recognition models?'.

After analysis of both the properties of the annotated data and the performance of the models, no notable difference was found in the quality of the training sets annotated by either author. The models trained on the data annotated by the AI student performed slightly better, but the margin was small. The most apparent reason for all results being this close together was the limited size of the training data. In validating the performance of the models, some entities were present only once or twice, or not at all. This leads to a distorted view of the performance of the model and so the difference in F1 score between models cannot be considered completely representative for the model qualities.

In addition to this, the decision to extract certain entities also proved to compromise the performance of the models the entity level as well. One example of this are entities that were not annotated 'consistently' or entities that were only recognized in 'duos'. Another example of a single entity compromising performance is the 'circumstances' entity, especially in the POL models. This entity proved to be of such length and variability, while occurring the most of all entities, that it was nearly impossible to generalize for the model.

However, the models trained on data annotated by the AI student and data annotated by the Police performed the similarly on this entity, even though the entity was annotated three times as little by the AI student. This suggests that having knowledge of NER itself is a more valuable asset than having knowledge of the topic of the datasets.

Overall, the decision to include the entities discussed above made this specific dataset especially difficult for a custom NER model to train on. Yet, these entities are of the most importance to the police. From this, it can be concluded that for some domains, creating a well-performing custom NER model seems to be a trade-off between extracting specifically the entities that are desired, but with a relatively poor performing model, or extracting more 'general' versions of those entities and create a model that might perform better. In either case, before any definite conclusions can be drawn on the importance of domain-specific expertise, training and testing the models on enough data is an even more essential preposition for training a well-performing custom NER.

References

[1] URL: https://allennlp.org/.

[2] Kai-Wei Chang, Vinod Prabhakaran, and Vicente Ordonez. "Bias and Fairness in Natural Language Processing". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): Tutorial Abstracts*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019. URL: https://www.aclweb.org/anthology/D19-2004.

[3] Michele Chinosi and Yaniv Steiner. *EMM Newsbrief*. URL: https://emm.newsbrief.eu/overview.html.

[4] *Data amp; informatieoplossingen*. URL: https://www.lexisnexis.nl/.

[5] *DeepL Translator*. URL: https://www.deepl.com/translator.

[6] Ralph Grishman and Beth M Sundheim. "Message understanding conference-6: A brief history". In: *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*. 1996.

[7] Martin Jurafsky. *Speech and Language Processing*. Pearson, 2008.

[8] Kenton Lee et al. "End-to-end neural coreference resolution". In: *arXiv preprint arXiv:1707.07045* (2017).

[9] Elske Nijhof. "Informatie Extractie uit Cocaïne gerelateerde Nieuwsartikelen". Scriptie HU. Jan. 2021.

[10] Joel Nothman, Tara Murphy, and James R Curran. "Analysing Wikipedia and gold-standard corpora for NER training". In: *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*. 2009, pp. 612–620.

[11] C. J. Van Rijsbergen. *Information Retrieval*. 2nd. Butterworth-Heinemann, 1979.

[12] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 3rd ed. Prentice Hall, 2010.

[13] Jingbo Shang et al. "Learning named entity tagger using domain-specific dictionary". In: *arXiv preprint arXiv:1809.03599* (2018).

[14] Hemlata Shelar et al. "Named Entity Recognition Approaches and Their Comparison for Custom NER Model". In: *Science & Technology Libraries* 39.3 (2020), pp. 324–337.

[15] Tony Sun et al. "Mitigating gender bias in natural language processing: Literature review". In: *arXiv preprint arXiv:1906.08976* (2019).

[16] *Training Pipelines Models · spaCy Usage Documentation*. URL: https://spacy.io/usage/training.

[17] Imed Zitouni. *Natural language processing of semitic languages*. Springer, 2014.