

MRP1 - GROUP 18 - FINAL REPORT

---

# **AUTOMATICALLY IDENTIFYING LEGAL CONCEPTS IN COURT DECISION TEXTS**

---

January 19, 2022

Department of Data Science and Knowledge Engineering (DKE)  
Supervisors: Dr. Kody Moodley, Dr. Rohan Nanda  
Coordinator: Dr. Gijs Schoenmakers

Jelle Jansen (6247224)  
Lucas Thil (6194698)  
Shishir Jain (6253834)  
Jan Schmidt (6079633)

# Contents

|          |                                                                      |           |
|----------|----------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                                  | <b>4</b>  |
| <b>2</b> | <b>Related Work</b>                                                  | <b>5</b>  |
| 2.1      | Dependency parsers . . . . .                                         | 5         |
| 2.2      | Deep learning approaches . . . . .                                   | 5         |
| 2.2.1    | Named Entity Recognition . . . . .                                   | 6         |
| 2.2.2    | Relation Extraction . . . . .                                        | 6         |
| 2.3      | Multi-headed output . . . . .                                        | 7         |
| <b>3</b> | <b>Approach</b>                                                      | <b>7</b>  |
| 3.1      | Dependency parser . . . . .                                          | 8         |
| 3.2      | Named Entity Recognition . . . . .                                   | 8         |
| 3.3      | Relations Extraction . . . . .                                       | 9         |
| 3.4      | Combining Named Entity Recognition and Relation Extraction . . . . . | 9         |
| <b>4</b> | <b>Data Set</b>                                                      | <b>9</b>  |
| 4.1      | Dataset Preprocessing . . . . .                                      | 10        |
| 4.2      | Dataset Annotations . . . . .                                        | 10        |
| 4.2.1    | Named Entities Annotations . . . . .                                 | 10        |
| 4.2.2    | Relations Annotations . . . . .                                      | 13        |
| 4.2.3    | Co-reference . . . . .                                               | 14        |
| 4.3      | Verification . . . . .                                               | 14        |
| <b>5</b> | <b>Model</b>                                                         | <b>14</b> |
| 5.1      | Dependency parsers . . . . .                                         | 14        |
| 5.2      | Named Entity Recognition . . . . .                                   | 15        |
| 5.3      | Relation Extraction . . . . .                                        | 15        |
| 5.4      | Two Headed Model . . . . .                                           | 17        |
| <b>6</b> | <b>Results</b>                                                       | <b>18</b> |
| 6.1      | Dependency parser . . . . .                                          | 18        |
| 6.2      | Named Entity Recognition . . . . .                                   | 19        |
| 6.3      | Relation Extraction . . . . .                                        | 20        |
| 6.4      | Two-Headed Bert Model . . . . .                                      | 21        |
| 6.5      | Examples . . . . .                                                   | 22        |
| <b>7</b> | <b>Conclusion and Discussion</b>                                     | <b>23</b> |

## Abstract

While there are many repositories available online for legal documents, its utilization i.e. identifying and extracting relevant information from this huge stack of unstructured information remains a challenge in the legal domain. It becomes essential that this challenge is addressed to capitalize on the available legal resources. Thus, our objective in this project is to identify and extract the entities and relations, such that important legal concepts are captured with the help of Natural Language Processing. Since deep learning approaches have been used, an annotated dataset needed to be created using raw text from the HUDOC database. This has been performed for both entities and relations, by combining manual, rule-based, and semi-supervised approaches. The task of identifying entities and extracting relations has been achieved via two steps, Named Entity Recognition and Relation Extraction. In this project, pre-existing models on the legal domain have been created as benchmark models. After that, a BERT model with two output heads has been implemented to perform both NER and RE, which is a first in the legal domain (according to our findings). The Two-Headed BERT Model performance was significantly higher than the benchmark models performance, with an accuracy of 96.25% and 95.96% for Named Entity Recognition and Relation Extraction respectively.

# 1 Introduction

In recent years, EU law has become increasingly digitized and published in textual documents on the Web in various repositories. One of these repositories, HUDOC, provides access to around 20,000 legal documents in English about court cases by the European Court of Human Rights [12]. These repositories give access to a huge amount of legal concepts. Legalization and regulations are created by combining these legal concepts, therefore legal concepts can be seen as the building blocks of law. With a plethora of EU legal texts being made available on the internet, it is important that those legal resources are utilized to the fullest. In the most commonly used information extraction techniques, the systems designed to retrieve relevant information can be improved, with respect to recognizing the meaning within the text, especially in the legal domain. This is due to the fact that the common approaches are predominantly rule-based on that do not pay attention to the semantic meaning of the sentences of the document while retrieving. Due to such instances, a considerable amount of information is lost at the application stage.

While we now have digital access to textual legal documents containing legal concepts, automatically extracting information from these texts using computers is still a challenge. The meaning of the content in these documents remains implicit and undetectable by machines. In this project, we try to build a tool that can process the semantic meaning of the text, and use them to extract legal concepts within the text.

The main research question that will be answered is:

- How can we automatically identify legal concepts in court decision texts by using the semantics within a text?

However, to answer this question, sub-questions should be answered before. For example, which real-world objects, called Named Entities (explained later on), are important in these documents? These Named Entities should be extracted from the textual documents. Besides that, we have to determine which relations provide higher coverage in terms of usage in these documents as well as carrying vital pieces of information between these Named Entities. In addition, we have an experiment which would be the most efficient way to extract these Named Entities and relations from a textual document. Thus the above questions can be rephrased and split into sub-questions.

1. Which entities and relations are important in court decision texts?
2. How do we automatically identify and extract entities and relations to identify legal concepts?

In order to extract these legal concepts, two steps are performed in sequence, first extracting of Named Entities (NE), and then extracting of Relations from the text by using the semantics within a text. Finally, both methods will get combined into one single step.

In Section 2, the related work and literature review will be discussed. After that, our work will be discussed, in multiple sections. Section 3, will talk about the Approach being used. Section 4, will talk about the creation of our dataset. Section 5 will describe the models designed and used. Once our work has been described, the results will be shown in Section 6. The conclusion and discussion of these results can be found in Section 7.

## 2 Related Work

A field that is specialized in information retrieval from text documents, is Natural Language Processing (NLP). It is to be acknowledged, that models in NLP are designed for specific domains. For example, an NLP model that performs flawlessly on text from the Medical domain is not guaranteed to perform well on Legal Text. Because of this, performance on different domains are hard to compare.

### 2.1 Dependency parsers

One of the common approaches to extract relations from Legal text is by using dependency parsers. For instance, take Reverb, which uses syntactical dependencies to extract relations within a sentence [9]. But other dependency parsers exists as well, for example ClausIE [6], Ollie [18], Stanford OpenIE [1], SpaCy dependency parser [22] and NLTK dependency parser [14]. Because the domain, on which the dependency parsers are being designed is important, there is also a specific one made for legal text, called LegOIE [20]. The advantage of these methods is that there is no need for training data since all rules have been designed by human experts. This allows the model to capture all relations, as long as all the rules have been defined. The disadvantage of this method is that there are a lot of exceptions in English grammar. Because of this, designing these parsers is a time-consuming task, and needs expertise in the language used (also consider the jargon used). But most importantly, the meaning of the text is not used when extracting the relations within the text. Only rule-based dependencies are considered.

### 2.2 Deep learning approaches

One of the most popular approaches to take the semantics into account as well is by using deep learning approaches, e.g. Bi-LSTM [11, 19] and BERT [7]. The models create embeddings (feature representations) for each word being used based on the context, by having connections to both the words that occurred before that specific word and the words that occurred after that specific word. These embeddings are also known as context embeddings.

However, using this method, extracting relations from text documents can be divided into two tasks, Named Entity Recognition (NER) and Relation Extraction (RE).

As the name of NER suggests, NER extracts Named Entities (NE) from the text. NEs can be seen as persons, locations, dates, etc. Using these NEs, relations can be extracted from documents in the Subject Verb Object form. Both the subject and the object will be NEs, where the verb is a relation between these two NEs, e.g. birth date, spouse, etc. For an visualised example see Figure 1.

Ousted **WeWork** founder **Adam Neumann** lists his **Manhattan** penthouse for **\$37.5 million**  
[organization] [person] [location] [monetary value]

(a) Named Entity Recognition

subject verb object  
**John ran to the shops.**

(b) Relation extraction

Figure 1: Examples Named Entity Recognition and Relation Extracting

### 2.2.1 Named Entity Recognition

Even though NER is a really known task, and has a high relevance, there is not a lot of research about it in the legal domain. Leitner et al. solved this, by comparing different methods on performance [13]. Different statistical approaches and deep learning approaches have been compared on German legal text. Deep learning approaches outperformed the statistical approaches. Their Bi-LSTM resulted in an accuracy of 95.95%.

Despite Leitner et al. comparing multiple statistical and deep learning approaches, they did not implement a BERT model. BERT works similar to a Bi-LSTM but uses the Attention Mechanisms to generally result in even higher performance. The state-of-the-art BERT model has been implemented in many domains for many NLP tasks. In this project, the model will be implemented to see if BERT can improve the performance of NER on the legal domain as well.

### 2.2.2 Relation Extraction

Once NER has been performed, RE can be used to extract the relation from the sentences. One method that is commonly used, can be seen in Figure 2 [17].

The pipeline consists of a classifier that classifies a single sentence into a relation within this sentence. Also for this pipeline, even though its high relevance, there is not a lot of research about it in the legal domain. A classifier that has been used is a Bi-LSTM, which should result in an accuracy of around 70% [17]. It should be noted, that as opposed to dependency parsers, only the predetermined relations can be classified using this approach. Because of this,

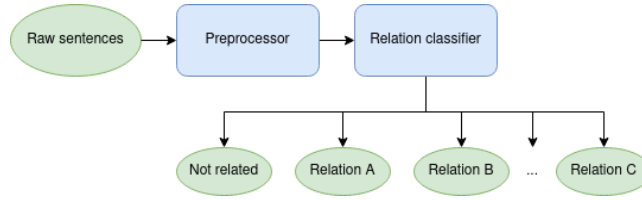


Figure 2: Classifying relations

it is important to determine these relations correctly. Besides that, even though semantics within a text get taken into account, deep learning approaches tend to be bad at recognizing negations. For example, for the sentence "Mr. X did not claim . . .", deep learning approaches consider it hard to recognize this negation. The model results in classifying this sentence with "claim". This should be kept in mind during the project. However, the advantage of this method is that it is easy to convert these relations into Subject Verb Object form [5].

### 2.3 Multi-headed output

Besides having two different steps for NER and RE, Xue et al showed that it is also possible to perform NER and RE at the same time on the Chinese Medical domain, using a Two-Headed Bert Model [24]. By using two output heads, the model can learn for RE, based on what it is learning for NER, and the other way around. This allows the model to perform even better. See Figure 3 for a visualisation.

The accuracy of this two-headed model on the Chinese Medical Documents is 89.39%. Because the Chinese Language is different from the English Language and it has been used on a different domain, it is interesting to see if this architecture also performs well on the English legal documents.

The advantage of this method is that the model can learn for RE, based on what it is learning, for NER, which is a much simpler task.

Since training these models can take a long time, pre-trained models on the same domain has been used. Chalkidis et al. created a pre-trained model, that was trained on the same dataset, see Section [4], with more computational power. That is why this model is used as an initial model before training since it already learned the Language Model.

## 3 Approach

In the previous section, multiple models and approaches have been mentioned. Since, in this project, the focus is on the semantic within the text, deep learning approaches will be the main focus. In this section, the approach, used during this project, will be discussed. First, the approach of building a benchmark model will be discussed. After that, building the final models will be discussed.

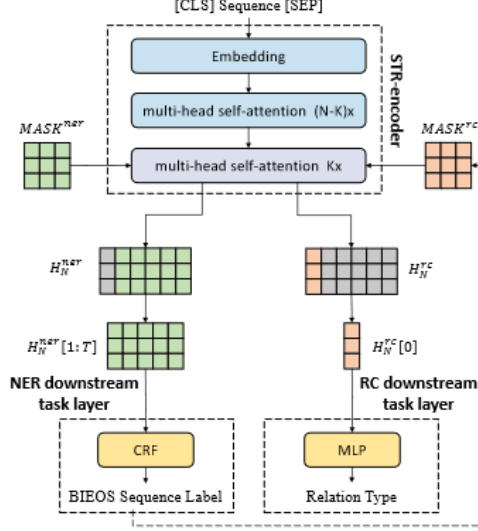


Figure 3: Two Headed model

### 3.1 Dependency parser

In Section 2.1, it is mentioned that dependency parsers fail to take the semantics into consideration. Besides that, a lot of rules have to be written to cover all exceptions within the sentence. To see if this is true, a pre-existing implementation, using the SpaCy framework, is used [15]. The implementation will show that it performs decently on short and factual sentences, but perform worse on sentences from a different domain, even after making adaptation to fit the domain of the text better.

### 3.2 Named Entity Recognition

According to Leitner et al. deep learning approaches, especially a Bi-LSTM, will perform best when performing NER on legal text. Since, in this project, the goal is to take the semantics of the text into account. Only deep learning approaches are being used.

For the initial model, a simple model has been implemented, because of the limited time, by only using a Bi-directional LSTM. Different sizes for the layers, nodes, and embeddings have been used, to see which sizes perform best. This has been performed on a part of the dataset only (around 20%), because of the long training times. The model is only used as a benchmark.

The model is created from the ground up. This means that the model has not been trained on any data before. Besides that, the tokenizer for this model has to be designed as well. To keep it simple, word tokenization has been used.

Word tokenization simply has a single token for every single word. So, "home" and "house" will both get a different one-hot encoding.

Besides selecting the tokenizer, there are several approaches to label output, IO, BIO, and BILOU are the most used ones. These letters indicate the boundaries of the entity being classified. This allows the model to identify the scope of a single Named Entity. B stands for Begin, I for Inside, L for Last, O for Outside, and U for Unit.

### 3.3 Relations Extraction

Using these NEs, relations can be extracted from the sentences as well. Also for this task, a deep learning approach is used as well. The same will be done as done for NER, only the output layer will be changed. So, the same architecture (Bi-LSTM), experiments, tokenization, etc., are performed. This model will be used as a benchmark model.

### 3.4 Combining Named Entity Recognition and Relation Extraction

Finally, when the experiments on these models have been performed and a benchmark has been created, both models have been combined into a single two-headed model and the performances have been compared.

This makes it able to compare the results for NER and RE on legal documents between Bi-LSTM and BERT. In contradiction to the previous models, this model is made more complex, inspired by the multi-headed Bert model on the Chinese Medical domain from Xue et al. [24]. The model designed contains two-output heads, one for NER and one for RE. Because of the limited time, a pre-trained legal BERT model on the HUDOC dataset (as described in Section 2.3) is being used. Because of this, the same tokenization has to be used. Unlike the benchmark models, which use word tokenization, the pretrained model uses WordPiece tokenization. WordPiece tokenization is generally used in BERT models. This tokenization splits words up into parts, based on the occurrences, otherwise, the word is kept in one piece. By using this, the model can learn meaning between parts of words. For example, "ex-girlfriend" and "ex-wife", both contain "ex" in which this part of the word has a specific meaning.

Also here, different configurations (e.g. sizes, parameters, etc.) have been used to see which model performs best.

## 4 Data Set

For the deep learning approaches an annotated dataset is required. During the research, it has been found that the legal data available online is mostly unlabelled. One amongst the few available labeled legal data was Contract Understanding Atticus Dataset (CUAD) [10]. Another paper that discussed labeling legal data was from Extracting Contract Elements [3]. It has to be

noted that these labeled datasets are restricted in their subject matter range. Thus, it was decided to utilize the HUDOC database for legal documents which deal with a wide range of subject matter. Although this meant that a huge dataset had to be annotated, which was a major task itself. To begin with, the dataset was obtained by scraping the HUDOC database which contains about 20,000 legal documents.

## 4.1 Dataset Preprocessing

To start off working on the dataset a few pre-processing steps have been incorporated to make the files usable. It is to be noted that about 40% of the available legal documents in the HUDOC database are corrupted. As such only 12,000 documents can be used. These documents are saved in the .docx file format, which is a problematic file to work with as such it is transformed into a .txt file and cleaned of all unusable characters. To annotate the document the sentences of the file are spitted and put into a line. This is done by going through each line of the documents and splitting them at the necessary points.

## 4.2 Dataset Annotations

The data obtained from the HUDOC database is unlabelled data. To address this major drawback, we had to find ways of annotating the vast dataset. In this project, multiple ways of annotating the legal dataset have been explored like manual annotation, semi-supervised annotating, and active learning. As such the documents have to be annotated. Two elements are necessary to be annotated:

- **Entity Annotation** is one of the most common processes in the generation of training datasets for NLP training tasks. It is the act of locating, extracting, and tagging entities in text. The type of entity annotation done in this project is for the use of NER.
- With **Relationship Annotation**, various relationships between different parts of your document are drawn.

### 4.2.1 Named Entities Annotations

For annotating the data, an automatic process has been used through combining SpaCy, manual annotation, and semi-supervised. Besides the labels used by SpaCy, multiple new entity labels have been added using regular expressions. SpaCy has been used, since, after some testing on the legal text, SpaCy performed best on the testset.

A total of 29 different labels for NEs were used. The commonly used labels that were implemented can be seen in Table 1

It is impossible to label all Named Entities individually, since there are infinite many, each domain has its own domain specific entity labels. Inspired by Dozier et al. and Leitner et al, additional labels have been added to the

| No. | Entity        | Description                                                                         |
|-----|---------------|-------------------------------------------------------------------------------------|
| 1   | COURT         | Juridical entity                                                                    |
| 2   | CASE          | ID and titles                                                                       |
| 3   | JUDGE         | Entity judging the case owning legal authority                                      |
| 4   | REGISTRAR     | The registrar for the board of judges                                               |
| 5   | APPLICATION   | The case application number                                                         |
| 6   | ARTICLE       | Part of a law text                                                                  |
| 7   | SECTION       | Refers to a part of a legal document                                                |
| 8   | PARAGRAPH     | Refers to a paragraph linked to a concept idea following general writing procedures |
| 9   | PROTOCOL      | Protocol                                                                            |
| 10  | VICTIMS       | Case's party seeking reparations                                                    |
| 11  | INVESTIGATORS | Party building the case                                                             |
| 12  | STATEMENTS    | A Party's declaration related to a section of the case                              |
| 13  | SECRETARY     | Assistant                                                                           |
| 14  | LAWYER        | Lawyers that the applicants have employed to present their case                     |
| 15  | DEFENDANT     | The entity the case is fought against                                               |
| 16  | PROSECUTOR    | The person leading the case against a defendant on the side of the state            |
| 17  | PETITION      | Petition or request                                                                 |

Table 1: SpaCy entities

common labels used [8, 13]. The law specific labels that were implemented can be seen in Table 2.

There are different ways to annotate documents. Manual annotation is the most straight forward annotation method possible. It has a high quality as a trained human has a good understanding of the necessary concepts needed to the labels needed. But this is a labor-intensive way of annotating unlabeled documents. A well-trained person needs at least 20 minutes per document. The only way to increase the speed of this annotation method can be done by increasing the number of people annotating the documents. This is a costly and still time-consuming method, as all persons annotating have to be trained to reach a certain amount of consistent quality. In the boundaries of the project, this is not feasible. A faster method is needed.

To improve the speed of annotations, a more automated method is needed. Semi-Supervised Annotations derive their ideas from semi-supervised learning. Semi-supervised learning is a learning paradigm concerned with the study of how computers and natural systems such as humans learn in the presence of both labeled and unlabeled data. The goal of semi-supervised learning is to understand how combining labeled and unlabeled data may change the learning behavior, and design algorithms that take advantage of such a combination. There are different methods for semi-supervised annotations: For Distant Learning, two

| No. | Entity        | Description                                                                         |
|-----|---------------|-------------------------------------------------------------------------------------|
| 1   | COURT         | Juridical entity                                                                    |
| 2   | CASE          | ID and titles                                                                       |
| 3   | JUDGE         | Entity judging the case owning legal authority                                      |
| 4   | REGISTRAR     | The registrar for the board of judges                                               |
| 5   | APPLICATION   | The case application number                                                         |
| 6   | ARTICLE       | Part of a law text                                                                  |
| 7   | SECTION       | Refers to a part of a legal document                                                |
| 8   | PARAGRAPH     | Refers to a paragraph linked to a concept idea following general writing procedures |
| 9   | PROTOCOL      | Protocol                                                                            |
| 10  | VICTIMS       | Case's party seeking reparations                                                    |
| 11  | INVESTIGATORS | Party building the case                                                             |
| 12  | STATEMENTS    | A Party's declaration related to a section of the case                              |
| 13  | SECRETARY     | Assistant                                                                           |
| 14  | LAWYER        | Lawyers that the applicants have employed to present their case                     |
| 15  | DEFENDANT     | The entity the case is fought against                                               |
| 16  | PROSECUTOR    | The person leading the case against a defendant on the side of the state            |
| 17  | PETITION      | Petition or request                                                                 |

Table 2: Custom entities defined for legal domain

different methods have been used to by utilizing a pre-trained model and a self build generic word detection.

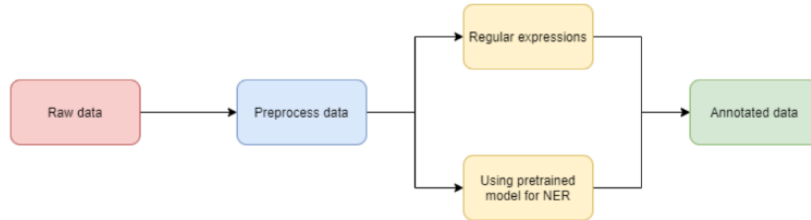


Figure 4: Annotation structure for Named Entity Recognition

Using a pre-trained model is useful for annotating common NEs found in lots of different texts. Due to using a SpaCy model for the pre-trained model annotation it is weak against noises in the training data, which can lead to flaws in the trainset for the models. To get a base set of annotations this is fine but for more specific entities (for which the model is not trained) this is not enough. For this, the second layer of annotation is used. For this Regular Expressions

are used. With the help of regular expressions, a lot of named entity candidates can be detected simply. Just by using common structures of the wording of the documents certain entities can be defined.

For example, most case names have a "v." in the middle or the applications are often described as "application", "no." and "[1-9]+/[1-9]2". With this method more of the law text, specific NEs can be labeled.

#### 4.2.2 Relations Annotations

Annotating relations is harder, because it indicates a dependency between two NEs. Since annotating by hand is a tedious task, a dependency parser and semi-supervised learning is used.

Snorkel, a semi-supervised learning method has been employed to annotate the relations within the dataset. Snorkel is a system for programmatically building and managing training datasets without manual labeling as described in [16]. In Snorkel, users can develop large training datasets in hours or days rather than hand-labeling them over weeks or months. This is done with heuristic rules or distant supervision techniques. By using Machine Learning, the model learns to label the data even better [21]. However, to build these heuristic rules, keywords from relations should be entered manually (e.g. the relation "sentenced" is described by possible keywords such as "sentenced", "penalized", "punished" etc).

Since there exist innumerable different relations between entities within the text, clausIE has been used to extract relations on a subset of documents. ClausIE, a dependency parser, has been used as this approach can recognize all relations in these documents. By using a word-counter, the predicates were ordered on the basis of their number of occurrences in the documents. By using this approach the coverage of the relations is high. A list has been created with the most occurring relations (after deleting stopwords).

A non-exhaustive list from the clausIE is: {decided, cited, ratified, dismissed, granted, complained, issued, submitted, adopted, apply, ordered, entitled, alleged, constituted, referred, claimed, ...}.

All of these keywords are quite commonly used in the legal domain and to an extent constitute legal jargon. This confirms that the relations that are extracted refer to legal concepts and that they can be identified. By combining this list, together with relations found manually, it was possible to create the heuristic rules for Snorkel. In our Snorkel implementation, 75 relations have been defined. These relations are listed in the Table 9 in the Appendix Section. Although these relations form a subset of all possible relations found in the legal text documents, it is to be noted that innumerable such relations can be defined in the implementation to capture specific relations between entities.

It is observed that this approach comes with its drawbacks. One is that the labels can only be labeled if the entities have been classified correctly. In addition, the sentence can only be labeled with a single relation which limits the expressiveness of the model. However, given the limited time, this approach for annotating the relations is justifiable.

### 4.2.3 Co-reference

Co-reference resolution is the task of finding all expressions that refer to the same entity in a text. It is an important step for a lot of higher-level NLP tasks that involve natural language understanding such as document summarizing. This can help in improving the overall quality of the annotations as entities referred via a co-reference are missed otherwise, because of long dependencies within the text. Implementing the necessary libraries has shown to be too time-consuming as it has lots of environment and versions problems. Besides that, performing this tasks on the documents is a time-consuming task, since it needs more than 20 minutes to run a single document. Since many documents have to be ran, it will take a considerable amount of time, which is not available because of the limited time for this project. Because of this, even though co-reference resolution is important to capture more information, it has not been used.

## 4.3 Verification

To validate the annotations made by the semi-supervised approach, the accuracies of the annotations have been calculated by annotating around 150 paragraphs of the dataset manual. This has been done by all group members. The accuracies are 0.95 and 0.72, with Cohen’s kappa values of 0.94 and 0.82 for NER and RE respectively. Even though these results are not perfect, the annotations are good enough to be used as training data.

# 5 Model

In this section the model designed will be explained, together with a visualization of the pipeline. The annotated data, described in Section 4 is used to train these models. In this section, first, the benchmark models for NER and RE will be described. After that, the combined model will be discussed.

## 5.1 Dependency parsers

To start, a dependency parser is being used to extract relations. The dependency parser is built, using the SpaCy library since this model considers the NEs when creating dependencies. This means that the model is partly using the semantics within the text.

To convert these dependencies into triples a pre-existing implementation is used [15]. Only the rules used (to create the triples), have been changed to suit the legal documents better. For example, everything before the relation tags ("ROOT", "adj", "attr", "agent", "amod") (UniversalDependencies [23]), is a subject, everything this is a object.

## 5.2 Named Entity Recognition

As described in Section 3.2 a Bi-LSTM is used. It uses one-hot encoding as an input, followed by an embedding layer, the Bidirectional LSTM (Bi-LSTM) layers, and a one-hot encoded output. Since the output contained BIO tags (described in Section 3.2), the model contained multiple outputs for the "same" tag (e.g. B-PERS, I-PERS, etc.). For a visualization of this model see Figure 5.

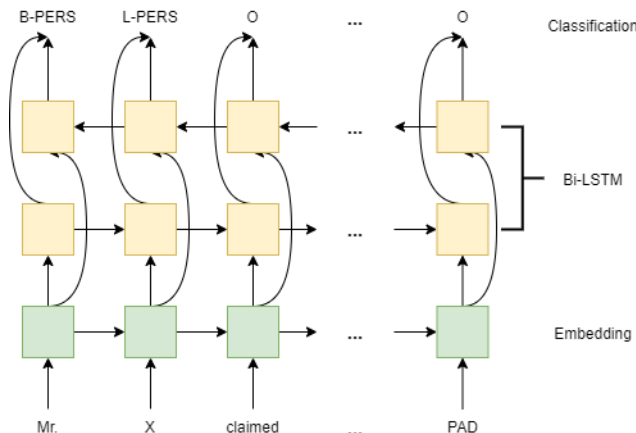


Figure 5: Named Entity Recognition Bi-LSTM

For this model, the tokenizer was designed ourselves. Because of this, a word tokenizer was used. This has been done, by creating one-hot encoding for every individual word in the trainset, and one token for Out of Vocabulary words.

The model got trained by using the Cross-Entropy loss, since the model is expected to have a single class output, together with an Adam Optimizer.

## 5.3 Relation Extraction

Also for this task, a Bi-LSTM is designed. The model is the same as seen for NER. The only difference is the output. Instead of having Sequence Labeling (a label for each input), a single output is needed. See Figure 6 for a visualisation.

Also for this model, the output is a one-hot encoding for all inputs. It is to be noted that with this design, only a single output/classification can be performed. This could be changed by creating an encoder, and a decoder. However, since there was not enough time to experiment with this, we will not go further into this. Especially since creating a dataset for this will take a considerable amount of time.

One thing that should be kept in mind, is that the data is highly imbalanced, see Figure 7. This could result in a biased model (for the majority class). Since this is a benchmark model only, the data got undersampled to balance the dataset [2].

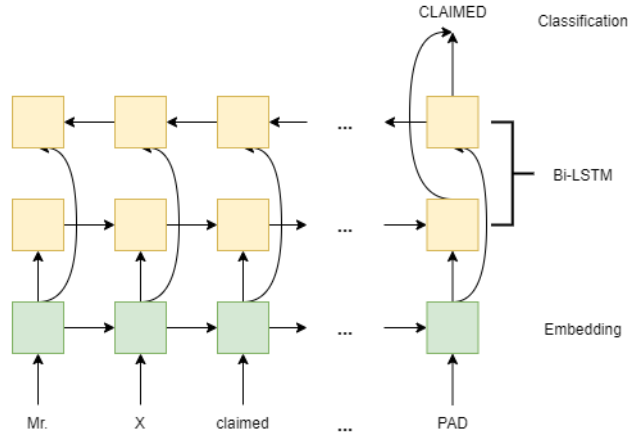


Figure 6: Relation Extraction Bi-LSTM

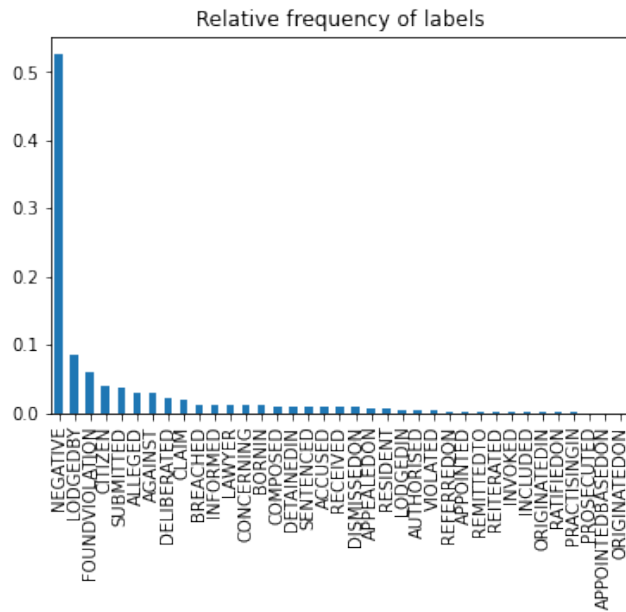


Figure 7: Imbalanced dataset

To determine SVOs, the output from the NER model, need to be combined with the output from this model.

## 5.4 Two Headed Model

Since both models are similar in architecture, the idea came up to combine them both in a single model (inspired by [24]). This allows the model to learn things for RE when updating its weights for NER, and the other way around. This could result in higher performance for both NER and RE.

Instead of using a Bi-LSTM again, a more complex model was used, BERT. Because of this, a comparison can be made between the performance of the Bi-LSTM and BERT. However, since BERT models are big, a pre-trained model, LegalBERT, was used at the core of the model [4]. This model only contains the Language Model of the entire HUDOC dataset but is not trained on NER or RE. However, since a pre-trained model is being used, the same tokenizer has to be used. In comparison to a word tokenizer (described in the previous models), a WordPiece tokenizer has been used (as described in Section 3.2).

The output of this pre-trained model is being used as an input for the two output heads, called the Two-Headed Bert Model (THBM). The outputs of these models are one hot encoded here as well. See Figure 8 for a visualisation of this model.

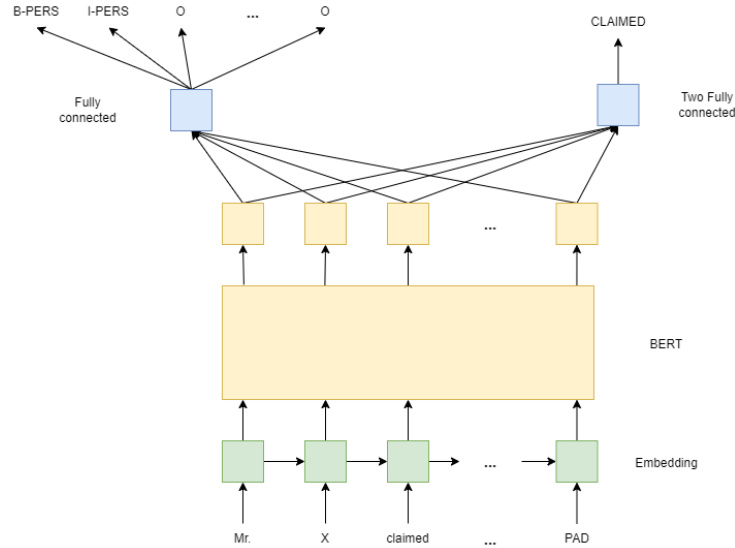


Figure 8: Two-Headed Bert Model

The model has been trained using the Cross-Entropy loss and Stochastic Gradient Descent optimizer for three epochs, where all weights (including the weights from the pre-trained model), were allowed to update.

By combining these two outputs, the SVOs can be found. Also here a decoder could be used to predict the NEs or relations (to allow multiple outputs). However, there was not enough time to implement and test this.

It has to be mentioned that this dataset is unbalanced, because of the labels

for RE. However, instead of undersampling the dataset, which is used for the benchmark RE model, a weighted sampler is being used. A weighted sampler, samples from the dataset (with replacement), such that minority classes occur more during training. This results in the model seeing all labels equally during training. Because of this, the model will not get biased towards a specific label.

## 6 Results

In this section, the results from the models will be discussed. The models have been trained as described in Section 5. First, the disadvantage of dependency parsers will be shown, after that the benchmark model of NER and RE will be discussed, at last, the Two-Headed model will be discussed.

### 6.1 Dependency parser

First, a dependency parser pipeline was created to extract SVOs, using the dependency parser from SpaCy. To visualize the triples, a visualization has been made, see Figure 9.

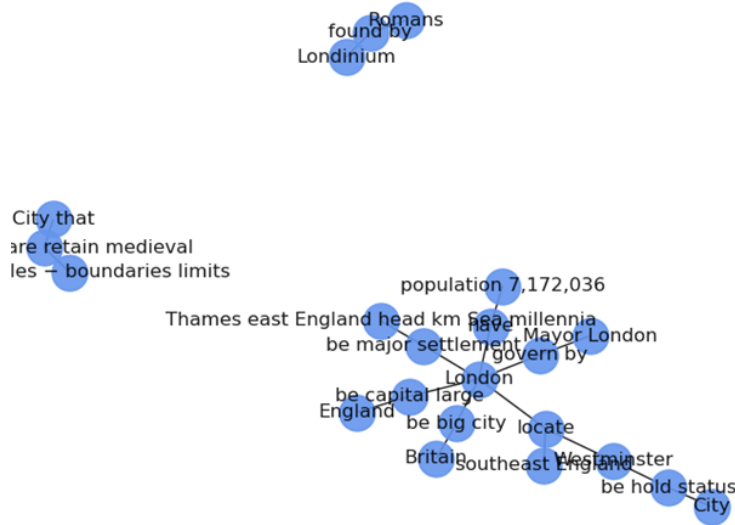


Figure 9: Dependency parser on factual sentences

The dependency parser performs decently on factual sentences but performed poorly on Legal texts, see Figure 10. Even after fine-tuning the rules, to be better suited for the Legal domain, there were a lot of exceptions. Because of the poor performance, no annotated dataset has been made for these triples, and no further has been done on dependency parsers within this project.

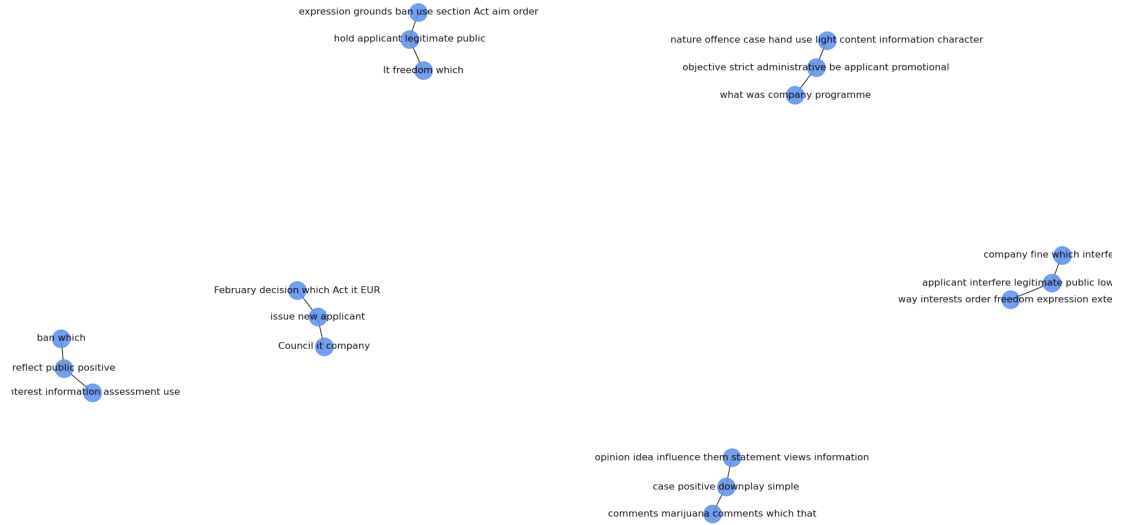


Figure 10: Dependency parser on Legal Texts

## 6.2 Named Entity Recognition

For NER, a Bi-LSTM was trained. After fine-tuning the parameters. The results are as follows:

| Model       | Measure  |           |        |        |
|-------------|----------|-----------|--------|--------|
|             | Accuracy | Precision | Recall | F1     |
| Bi-LSTM NER | 0.9510   | 0.9430    | 0.9510 | 0.9321 |

Table 3: Performance of benchmark model on NER

In Figure 11 the accuracy during training can be seen (blue line). The model does not seem to overfit, since the accuracy on the validation set (orange line) increases together with the train data. Besides that, the model seemed to converge, since the performance does not increase significantly anymore.

The parameters used can be seen in Table 4

| Parameter          | Value |
|--------------------|-------|
| Embedding size     | 51    |
| Hidden states LSTM | 102   |
| Fully connected    | 51    |
| Batch size         | 128   |
| Epochs             | 15    |

Table 4: Parameters used of benchmark model on NER

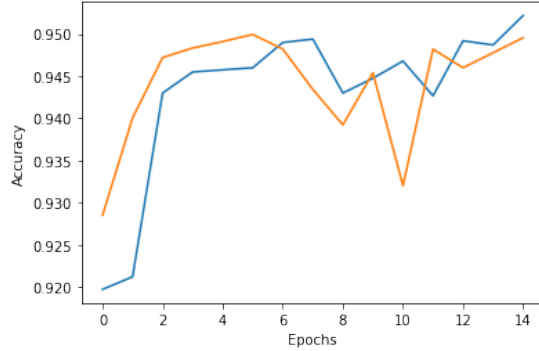


Figure 11: Accuracy during training benchmark model NER

Even though the model is relatively simple, the model performed well. The accuracy achieved is (almost) similar to the accuracy on the German Legal Texts (as seen in Section 2.2.1).

### 6.3 Relation Extraction

Similar to NER, a Bi-LSTM was trained, for RE as well. After fine-tuning the parameters. The results are as follows:

| Model      | Measure  |           |        |        |
|------------|----------|-----------|--------|--------|
|            | Accuracy | Precision | Recall | F1     |
| Bi-LSTM RE | 0.4659   | 0.6731    | 0.4659 | 0.5219 |

Table 5: Performance of benchmark model on RE

In Figure 12 the accuracy during training can be seen (blue line). The performance on the validation set does not increase (orange line), while the performance on the trainset increases. This overfitting can be caused, by the fact that undersampling was being used. The dataset did not have enough examples and resulted in an overfitted model.

The parameters used can be seen in Table 6.

| Parameter                 | Value |
|---------------------------|-------|
| Embedding size            | 256   |
| Hidden states LSTM        | 128   |
| Fully Connected Layer One | 128   |
| Fully Connected Layer Two | 64    |
| Batch size                | 32    |
| Epochs                    | 15    |

Table 6: Parameters used of benchmark model on RE

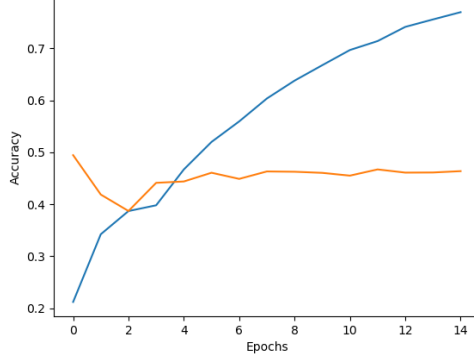


Figure 12: Accuracy during training benchmark model RE

The model does not seem to perform well as the accuracy is lower than the benchmark model discussed in Section 2.2.2. However, it is to be noted that this performance was measured on a completely different domain. One of the reasons for the reduced model could be due to the undersampling of the dataset, where the dataset could be too small to be trained properly.

#### 6.4 Two-Headed Bert Model

Lastly, THBM, a BERT model with only one output head, has been implemented to perform both NER and RE. Because of the relation labels, the dataset was unbalanced, as seen in Figure 7. A weighted sampler has been used to train the model since undersampling resulted in poor performance. The results are as follows:

The overall results, of all models, can be found in Table 7.

| Model    | Measure  |           |        |        |
|----------|----------|-----------|--------|--------|
|          | Accuracy | Precision | Recall | F1     |
| THBM NER | 0.9625   | 0.9502    | 0.9625 | 0.9525 |
| THBM RE  | 0.9596   | 0.9671    | 0.9596 | 0.9602 |

Table 7: Performance of Two-Headed Bert Model (NER and RE)

The THBM has the highest performance for both NER and especially RE. The Attention Mechanisms within the BERT model, improves the performances significantly, compared with the Bi-LSTM. Beside that, the performance is significantly higher than the model designed on the Chinese Medical domain (as seen in Section 2.3). The loss during training of this model can be seen in Figure 13. It seemed like the model converged to a point, since the loss isn't decreasing significantly.

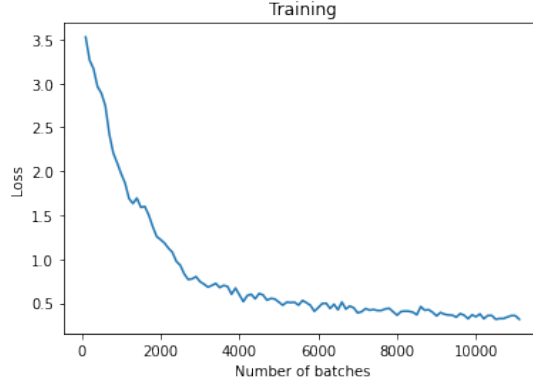


Figure 13: Loss during training Two-Headed Bert Model

One of the impressive aspects to be mentioned is that the model was able to recognize person names and company names correctly. Given a single sentence, it is a difficult task even for humans as there are company names from different languages that are hard to distinguish from a person's name. In addition, some documents referred to words from other languages (such as "Rijkswaterstaat" in a Dutch document), which the model was able to recognize successfully. In short, even the Out of Vocabulary words (OOV) were successfully predicted by the model.

During the implementation of the THBM, two separate BERT models with only one output head have been implemented for both NER and RE, during error shooting. After training these models, it was noticed that NER performed equally well when the second head was added in the THBM. Interestingly, the performance for RE increased significantly after the addition of the second head (in the THBM, from 70% to around 95%). This implies that the performance of RE increased because it was able to learn from NER.

## 6.5 Examples

Three examples of the THBM creating SVOs can be seen in Table 8.

As visible in Table 8, the model and pipeline are able to successfully create triples from sentences. However, some potential improvements are visible by looking at these examples. Example 1, received an (almost) correct triple. However, the scope of the object is not correct, the number "1" should also be added. Example 2, received a triple as well. But, since the model/pipeline did not learn how to handle co-references, the model failed to extract the subject correctly. The same applies to Example 3, where the model/pipeline failed to predict the "applicants", as the subject. Because of this, no relation was found (while the predicted relation/verb was correct).

Because of the limited time for this project, these improvements could not be made anymore, but are left for further research.

| Example | Sentence                                                                                                                                                                                                                                                                                                                                                | Triple                                                                                            |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| 1       | In this context , the Court notes that the origin of the violation found under Article 1 of Protocol No . 1 was the absence of compensation for such interference with the applicant companies property rights and not , as the applicant company puts it , the unlawful taking of its property ( see the principal judgment , § § 52 , 60 , and 61 ) . | <b>Subject:</b> court,<br><b>Verb:</b> FOUNDVIOLATION,<br><b>Object:</b> article 1 of protocol no |
| 2       | The applicant was represented by Mr Kostyantyn Buzadzhy , a lawyer practising in Kyiv .                                                                                                                                                                                                                                                                 | <b>Subject:</b> by,<br><b>Verb:</b> LAWYER,<br><b>Object:</b> kost ##yan ##ty                     |
| 3       | The applicants were born in 1963 and 1950 respectively and live in .                                                                                                                                                                                                                                                                                    | <b>Verb:</b> BORNIN,<br>No relations found                                                        |

Table 8: Examples of extracted triples

## 7 Conclusion and Discussion

By combining previous research and the findings during this project, important NEs have been successfully labeled using Regular expressions and a pre-trained model, by a semi-supervised approach. Besides the NEs, important relations have been found by using a manual approach combined with a pre-existing dependency parser. This resulted in detecting the common relations within the text, with high coverage. The benchmark model implemented for NER resulted in a similar performance on the HUDOC dataset (95.95% and 95.10%), compared to pre-existing research on other texts. It can be said that, the results from this project are comparable to the results of other research on legal domain. On the other hand, RE performance reduced, compared to existing research results (70% to 46.59%). This was caused by the fact that undersampling was used, which resulted in a too-small dataset. However, this has been solved in the THBM by using a weighted sampler to handle the imbalanced dataset. By doing so, NER performed slightly better (96.25%), simultaneously, it resulted in a significant improvement on the performance of RE (95.96%). Using two heads was a key to improving the performance of RE as it was able to learn from NER, which contributed to the increased performance. Looking at these results, it can be concluded that the annotation performed was successful enough to train models on this dataset. By using the output of both NER and RE, SVO triples were created successfully, using a rule based approach. These triples could be easily adopted to Knowledge graphs, by changing the labels for NER and RE accordingly. The implementation of this project can be found on Github <sup>1</sup>

The project has produced promising results, yet there is scope for further research and improvement. For the annotation of the dataset, a lot of im-

<sup>1</sup>[github.com/GitHubByJelle/NER\\_RE\\_HUDOC](https://github.com/GitHubByJelle/NER_RE_HUDOC)

provements could be made, especially errors made by the SpaCy model have a negative influence on the training data. The quality of the annotations can possibly be improved by manual annotation or Active Learning instead of the annotation method used in this project. As earlier discussed the pipeline fails to capture long relations within the text. This is caused by co-references used within the text. Because of the limited time, co-reference resolution has not been incorporated in this project. However, co-reference resolution has the potential to significantly improve the information retrieval from the legal text. Currently, the model is only able to handle single relations, but there are scenarios in which multiple named entities or relations are present in a single sentence. Hence, the annotation could be adapted to capture multiple named entity labels and multiple relations and accept multiple SVO annotations. This implies that the models should be changed as well. For example, an encoder-decoder architecture could be used. Finally, the current rule-based approach to create SVO could also be changed to pointer networks, to extract the SVO differently.

## References

- [1] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. “Leveraging linguistic structure for open domain information extraction”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2015, pp. 344–354.
- [2] Tara Boyle. *Dealing with Imbalanced Data*.
- [3] Ilias Chalkidis, Ion Androutsopoulos, and Achilleas Michos. “Extracting contract elements”. In: *Proceedings of the 16th edition of the International Conference on Artificial Intelligence and Law*. 2017, pp. 19–28.
- [4] Ilias Chalkidis et al. “LEGAL-BERT: The muppets straight out of law school”. In: *arXiv preprint arXiv:2010.02559* (2020).
- [5] David Crystal et al. *The cambridge encyclopedia*. Cambridge University Press Cambridge, 1990.
- [6] Luciano Del Corro and Rainer Gemulla. “Clausie: clause-based open information extraction”. In: *Proceedings of the 22nd international conference on World Wide Web*. 2013, pp. 355–366.
- [7] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].
- [8] Christopher Dozier et al. “Named Entity Recognition and Resolution in Legal Text”. In: *Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language*. Ed. by Enrico Francesconi et al. Vol. 6036. Lecture Notes in Computer Science. Springer, 2010, pp. 27–43. DOI: 10.1007/978-3-642-12837-0\\_2. URL: [https://doi.org/10.1007/978-3-642-12837-0\\\_2](https://doi.org/10.1007/978-3-642-12837-0\_2).
- [9] Anthony Fader, Stephen Soderland, and Oren Etzioni. “Identifying relations for open information extraction”. In: *Proceedings of the 2011 conference on empirical methods in natural language processing*. 2011, pp. 1535–1545.
- [10] Dan Hendrycks et al. “Cuad: An expert-annotated nlp dataset for legal contract review”. In: *arXiv preprint arXiv:2103.06268* (2021).
- [11] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-term Memory”. In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.
- [12] *Hudoc*. <https://www.echr.coe.int/>. Accessed: 2022-01-12.
- [13] Elena Leitner, Georg Rehm, and Julian Moreno-Schneider. “Fine-grained named entity recognition in legal documents”. In: *International Conference on Semantic Systems*. Springer. 2019, pp. 272–287.
- [14] *NLTK Dependency Parser*. <https://www.nltk.org/api/nltk.parse.html>, note = Accessed: 2022-01-12.

- [15] *Python-Knowledge-Graph*. <https://github.com/bdmarius/python-knowledge-graph/blob/master/knowledgegraph.py>. Accessed: 2022-01-17.
- [16] Alexander Ratner et al. “Snorkel: Rapid training data creation with weak supervision”. In: *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*. Vol. 11. 3. NIH Public Access. 2017, p. 269.
- [17] *Relation extraction with BERT*. <https://www.kaggle.com/duongthanhhung/bert-relation-extraction>, note = Accessed: 2022-01-12.
- [18] Michael Schmitz et al. “Open language learning for information extraction”. In: *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*. 2012, pp. 523–534.
- [19] Mike Schuster and Kuldeep Paliwal. “Bidirectional recurrent neural networks”. In: *Signal Processing, IEEE Transactions on* 45 (Dec. 1997), pp. 2673–2681. DOI: 10.1109/78.650093.
- [20] Giovanni Siragusa et al. “Relating Legal Entities via Open Information Extraction”. In: *Research Conference on Metadata and Semantics Research*. Springer. 2018, pp. 181–187.
- [21] Snorkel. *Programmatically Build Training Data*. <https://www.snorkel.org/>, note = Accessed: 2022-01-12.
- [22] *SpaCy Dependency Parser*. <https://spacy.io/api/dependencyparser>, note = Accessed: 2022-01-12.
- [23] *Universal Dependencies*. <https://universaldependencies.org/>, note = Accessed: 2022-01-12.
- [24] Zhichang Zhang, Lin Zhu, Peilin Yu, et al. “Multi-level representation learning for Chinese medical entity recognition: Model development and validation”. In: *JMIR medical informatics* 8.5 (2020), e17637.

## Appendix

| No. | Relationship Types | Relationships                                                            |
|-----|--------------------|--------------------------------------------------------------------------|
| 1   | DATE_COURT         | QUASHEDJUDGMENT, ACQUITTED, QUASHEDDECISION                              |
| 2   | PERSON_PERSON      | LAWYER, ACCUSED, ALLEGED, AUTHORIZED                                     |
| 3   | PERSON_DATE        | BORNIN, SENTENCED, APPEALEDON                                            |
| 4   | PERSON_GPE         | BORNIN, RESIDENT, CITIZEN, PRACTISINGIN, DETAINEDIN                      |
| 5   | PERSON_MONEY       | CLAIM, RECEIVED, SOUGHTREPARATION, SOUGHTREIMBURSEMENT                   |
| 6   | PERSON_ARTICLE     | VIOLATED, BREACHED, COMPLAINEDUNDER, ALLEGEDBREACH                       |
| 7   | PERSON_RULE        | VIOLATED, GRANTEDLEAVE, COMPLAINEDUNDER, ALLEGEDBREACH                   |
| 8   | PERSON_NORP        | LODGEDCOMPLAINTWITH, APPEALEDTO, LODGEDCOMPLAINTAGAINST, APPEALEDAGAINST |
| 9   | PERSON_ORG         | LODGEDCOMPLAINTWITH, APPEALEDTO, LODGEDCOMPLAINTAGAINST, APPEALEDAGAINST |
| 10  | PERSON_COURT       | APPEALEDTO, INFORMED, CHALLENGED-DECISION, SUBMITTED                     |
| 11  | ORG_PERSON         | COMPOSED, REPRESENTEDBY, APPOINTED, INFORMED                             |
| 12  | ORG_DATE           | BORNIN, DELIBERATED, ORIGINATES                                          |
| 13  | ORG_ARTICLE        | VIOLATED, BREACHED, APPOINTEDBASEDON                                     |
| 14  | ORG_RULE           | VIOLATED, BREACHED, APPOINTEDBASEDON                                     |
| 15  | COURT_PERSON       | CHAMBERCOMPOSITION, CONVICTED, INDICTMENTAGAINST                         |
| 16  | COURT_ARTICLE      | FOUNDVIOLATION, FOUNDNOVIOLATION, CITED                                  |
| 17  | COURT_PROTOCOL     | FOUNDVIOLATION, FOUNDNOVIOLATION, CITED, HOLDSVIOLATION                  |
| 18  | COURT_RULE         | FOUNDVIOLATION, FOUNDNOVIOLATION, CITED                                  |
| 19  | COURT_DATE         | ACCEDEDREQUEST, DISMISSEDON, RATIFIEDON                                  |
| 20  | NORP_RULE          | DECISIONUNDER, VIOLATED, APPOINTED-BASEDON                               |
| 21  | NORP_ARTICLE       | DECISIONUNDER, VIOLATED, APPOINTED-BASEDON                               |

| No. | Relationship Types  | Relationships                                      |
|-----|---------------------|----------------------------------------------------|
| 22  | NORP_PERSON         | CHAMBERCOMPOSITION, REPRESENT-<br>EDBY, APPOINTED  |
| 23  | APPLICATION_GPE     | AGAINST, FILEDIN, DISMISSEDIN                      |
| 24  | APPLICATION_PERSON  | LODGEDBY, LODGEDAGAINST, CONCERN-<br>ING           |
| 25  | APPLICATION_DATE    | LODGEDIN, SUBMITTEDON, RELIEDON                    |
| 26  | APPLICATION_ARTICLE | LODGEDUNDER, INVOKED, REITERATED                   |
| 27  | CASE_ORG            | REFERREDBY, REFERREDTO, FILEDA-<br>GAINST          |
| 28  | CASE_DATE           | REFERREDON, ORIGINATEDON, SETTLE-<br>DON           |
| 29  | CASE_COURT          | REFERREDTO, REMITTEDTO, SUBMIT-<br>TEDTO           |
| 30  | CASE_PERSON         | LODGEDBY, LODGEDAGAINST, FILEDA-<br>GAINST         |
| 31  | CASE_APPLICATION    | ORIGINATEDIN, REFERREDTO, INCLUDED                 |
| 32  | PROSECUTOR_PERSON   | INDICTMENTAGAINST, AUTHORISED, PROS-<br>ECUTED     |
| 33  | PROSECUTOR_ORG      | INDICTMENTAGAINST, AUTHORISED, PROS-<br>ECUTED     |
| 34  | PROSECUTOR_NORP     | INDICTMENTAGAINST, AUTHORISED, OR-<br>DEREDAGAINST |
| 35  | PROSECUTOR_COURT    | REQUESTED, NOTED, SUBMITTED                        |
| 36  | PROSECUTOR_DATE     | REQUESTEDON, SUBMITTED, PROSECUTE-<br>DON          |

Table 9: Table presenting the defined relationship types and the defined relationships under each relationship type