Utrecht University

Graduate School of Natural Sciences
Artificial Intelligence

Master Thesis
Submitted in partial fulfilment of the requirements for the degree of Master of Science

Extracting Structured Information from English Legislative Text

A Comparative Analysis of Language Models for FLINT Frame Role Labelling

By

A. Juul Schoevers

Student number: 6026923

University Supervisors:

M.P. Schraagen
F.J. Bex

TNO Supervisors:

R.M. Bakker
M.H.T. de Boer

July, 2023

Acknowledgements

I would like to express my deepest gratitude to my supervisors Marijn Schraagen and Roos Bakker, for their support and patient guidance during this thesis project. Both of them spent a significant amount of their time providing me with valuable insights and feedback, which stimulated me to get the most out of this process. Without their expertise and encouragement, this thesis would not have come about.

I would also like to thank Maaike de Boer and Romy van Drie for the engaging discussions and uplifting talks, as well as Ioannis Tolios for his help setting up the annotation process and patiently addressing all of my requests. Moreover, I extend my sincerest thanks to Fenna Blom, Lucas Snijder, Yulia Terzieva, and Daan van der Weijden for their invaluable contributions to this thesis project. Despite their own internship commitments, they generously devoted a substantial amount of time to annotating my data.

Finally, I would like to thank my family and friends for motivating me and for always making me have a good laugh.

Abstract

The long and complex nature of legal texts poses difficulties in interpreting these texts, especially for non-expert readers. To address this challenge, the Flint framework has been proposed to make the interpretations of normative texts, including legal texts, explicit by capturing them in act and fact frames. However, constructing Flint frames requires a lot of manual labour. This thesis focuses on automating part of the Flint frame-filling process by exploring a system that automatically labels legal texts with the semantic roles associated with the Flint act frame. This study builds upon previous work by Bakker et al. (2022), which uses rule-based and transformer-based models to label act frame roles in Dutch law text and extends their methodology to the English language.

First, an extensive annotation process was conducted, resulting in a dataset of 1539 action sentences from European Union regulations, annotated with Flint act frame roles. Subsequently, several models were implemented to label these action sentences with Flint roles. A rule-based baseline was developed, along with a model that maps standard semantic roles to Flint roles. Moreover, fine-tuned BERT models were implemented, including BERT, LEGAL-BERT, EURLEX-LEGAL-BERT, and SpanBERT. Lastly, a multilingual BERT model was fine-tuned on labelled sentences from a larger Dutch dataset to explore the potential of cross-lingual transfer learning for Flint role labelling.

The models were evaluated on the token level and, additionally, the MUC-5 metric was used to determine to what extent models can correctly identify roles as a whole. The findings demonstrated that fine-tuning models was the most effective approach and that domain-specific language models had an advantage over the other fine-tuned models. We conclude that fine-tuning a model to label Flint roles is a promising step toward automatically filling Flint frames.

Contents

1 Introduction

Human language is inherently ambiguous and, as a consequence, our speech and written texts are often open to more than one interpretation. Legal texts are no exception since this linguistic ambiguity is also present in the legal domain and, despite the care and attention with which legislative texts are drafted, these texts are often difficult to understand, especially for non-expert readers.

The long and complex nature of law texts makes it challenging to interpret them, search them for relevant information, compare them to other legislative texts, or understand the consequences for particular agents of a certain piece of law text. Moreover, law texts are not immune to having multiple interpretations. Having multiple interpretations of the law is not inherently a bad thing, but it can become problematic when conflicts of these interpretations arise. Such a conflict may occur in many situations, for example, while working in multidisciplinary teams, when judges with similar cases make different decisions, or when a law applies under circumstances that were not anticipated at the time that law was written.

Because of the ambiguous nature of natural language, it is unlikely that the aforementioned challenges can be avoided entirely. Therefore, it could be practical to have a method to create a formal interpretation of pieces of law text to help us overcome some of these challenges since it makes the text easier to digest for humans and machines. It could also help us understand the nature of a potential difference in the interpretation of the law, allowing for a better comparison of different interpretations, which could function as a first step in resolving conflicts.

In recent years, Flint has been proposed as a method for making the interpretations of sources of norms explicit (van Doesburg & van Engers, 2019). Here, norms are understood as guidelines that influence human behaviour. Flint is a formal knowledge representation language that formalises interpretations of norms in two different types of frames: act frames and fact frames (for a more detailed description of the Flint language, see section 2.2.3). The Flint frames provide an explicit overview of the interpretation of a norm and can be used as a basis for a reasoner that can use them to model agent behaviour in a normative setting or for conflict resolution later down the line. For example, the sentence "*Competent authorities shall report to the Commission a list of the contractual agreements approved.*" contains a norm that describes how certain actors (*competent authorities*) are expected to act (*report*) in relation to other agents (*the Commission*). This action can be formalised in a Flint act frame by extracting its essential components — such as the agents involved, the actions or obligations described, and pre- or post-conditions associated with the action — and organising them in their associated slots in the frame. This example pertains to a single sentence, but it should be noted that Flint can capture and formalise norms that go beyond single sentences.

One of the most prominent limitations of implementing this framework in real-

life settings is that constructing Flint frames requires a lot of labour. Law texts are usually large and complex bodies of text containing many different norms that demand expert knowledge to understand. Hence, manually creating Flint frames is a considerable task requiring a lot of time and domain knowledge to complete. A system that automatically fills Flint frames would lift much of this burden. Therefore, in this thesis, we explore a system that can automate a part of the process of automatically creating Flint frames from pieces of law text. More specifically, we will explore several methods to identify in sentences from English legal text the spans of text that correspond to the first four constituents of Flint act frames (henceforth Flint roles) which are the *actor*, *action*, *object*, and *recipient*. This task is comparable to the semantic role labelling task. Hence, we are trying to solve for each sentence: who (*actor*) does (*action*) what (*object*) to whom (*recipient*)?

Because of recent developments in Deep Learning (DL) and Natural Language Processing (NLP), these seem like promising areas for exploring such an automated system for labelling sentences with Flint roles. We have already seen several attempts to employ NLP techniques to formalise the interpretations of norms. Most notably, Bakker et al. (2022a) created a method using a transformer-based language model to label the constituents of Flint act frames in Dutch law text, yielding promising results. In this thesis, we will build upon this work as we focus on legal text in English, taking European Union (EU) regulations as a case study.

Our decision to focus on English legal text is motivated by the fact that English is commonly used in many different legal contexts, for example, in international and European law. By working on a model that can label English legal text with Flint roles, we aim to enable broader access to the Flint framework throughout the international legal community. Additionally, several resources offer language models tailored to analyse English legal text. This availability of English legal language models allows us to compare several of these models, thereby carrying out a more comprehensive analysis which we hope will result in a more valuable contribution to the field.

The decision to use EU regulations for our analysis is motivated by the fact that these dictate many actions that individuals and organisations can and cannot take. These laws contain many sentences that contain descriptions of actions, making them ideal candidates for labelling with the roles associated with the Flint framework's act frame.

This introduction discusses the challenge of automatically labelling Flint act frame roles in section 1.1 and the potential of using a transformer-based approach in section 1.2 before formulating our research questions in section 1.3. In section 1.4, we conclude this introduction with an overview of how this thesis is structured.

1.1 Automatically extracting Flint act frame roles

Automating the construction of Flint frames from legal text would allow for formalising law texts on a larger scale. To this end, this thesis aims to explore a method for automatically labelling sentences with the components that can be used to build Flint frames in sentences from several EU regulations. Note that we will not delve into the subsequent step of automatically filling the Flint frames after extracting the Flint roles from single sentences, as this process requires additional information. For example, apart from identifying the Flint roles in a sentence, this also requires un-

derstanding how these roles relate. This becomes increasingly complex as sentences contain multiple instances of the same role. Furthermore, filling a Flint act frame often requires information beyond the sentence level, given that information that should be contained in a single act frame slot is often fragmented and spans multiple sentences. For example, consider the two subsequent sentences "*The members of the Commission are legal experts.*" and "*They report to the competent authorities.*". Here, the action (*to report*) is contained in the second sentence, but we would not want to fill the associated *actor* slot with the word *they* from that same sentence. Rather, we would link *the members of the Commission* from the first sentence to this action. Another example is provided in appendix A.1, which contains a manually constructed Flint act frame for an article from the General Data Protection Regulation (GDPR). This shows that even though we might be able to identify a span of text that is associated with a slot in the act frame within a sentence, we can not necessarily fill in said slot with this sequence of text. Instead, we can use the roles from single sentences as building blocks for eventually filling the Flint frame. As such, the systems developed in this thesis are intended as a stepping stone to a complete system for the automatic filling of Flint frames.

In our work, we will only focus on the semantic roles that are a part of the Flint act frame and will not cover the fact frame of the Flint language. This decision was made to ensure an in-depth study for the act roles. We choose to focus on actions rather than facts because we can consider Flint, as a formal language for modelling normative text, to be centred around actions given that norms monitor the actions of agents. However, we expect that our methodology for labelling the semantic roles associated with the act frame can be adapted relatively easily for the Flint fact frame, which is a potential direction for future work.

The work by Bakker et al. (2022a), upon which the work in this thesis builds, only focuses on extracting the first four Flint act frame roles from individual sentences, because these form the core of the Flint act frame and bear the most similarity to more common semantic roles. For this reason, we also limit our focus on identifying these same four Flint roles within the confines of a sentence.

1.2 Transformer-based approaches

In recent years, the field of NLP has seen impressive leaps forward in developing transformer-based language models, which have achieved state-of-the-art results on a wide range of NLP tasks. The transformer architecture, which was introduced several years ago by Vaswani et al. (2017), effectively captures the relationships of words in pieces of text and has served as a basis for many recently developed language models, one of the most notable ones being BERT (Devlin et al., 2018). In this work, we will use several variations of the BERT language model for labelling legal text with Flint act roles.

We compare the effectiveness of three different approaches to label action sentences with the Flint act roles. Firstly, we implement a rule-based model as our baseline, based on the model by Bakker et al. (2022b). Second, we will use a BERT-based model trained on a semantic role labelling (SRL) task to find the semantic roles within the sentences. We will research how these SRL roles relate to Flint frame roles to create a mapping from SRL roles to Flint roles. Lastly, we will study the effectiveness of fine-tuning a BERT model on a dataset of sentences annotated

with Flint roles. When data is scarce, we can use a model pretrained on a large amount of data for a related task and leverage the knowledge from this model by fine-tuning the model for the target task with the smaller amount of data. This method has been responsible for many recent improvements in performance on NLP tasks. Given that there are no data available that identify acts or Flint roles in English text, we will provide an annotated dataset of 1539 sentences labelled with Flint act frame roles, enabling us to utilise the fine-tuning technique. To obtain this dataset, we conduct an extensive annotation process with five annotators, based on an elaborate annotation protocol. This dataset provides a valuable resource for the task in this thesis and, potentially, for future work in the legal information extraction domain.

With the annotated dataset, we fine-tune several language models with different properties potentially useful for our legal semantic role labelling task. We fine-tune the original BERT model and experiment with domain-specific BERT models tailored to the legal domain. Domain-specific language models have been trained on text specific to a particular domain, allowing these models to better capture the relations in texts from their target domain. We want to research the effectiveness of using a domain-specific model in the legal context since we hypothesise that using a domain-specific language model will allow us to obtain a better formal model of the legal text, ultimately yielding better results in identifying Flint act roles. We also fine-tune a BERT model pretrained to predict spans of text, as we are looking to identify spans of text (the semantic roles) in our semantic role labelling task. We expect that this model can lead to increased performance on our task, as it better captures the relations between words in a text and would therefore be better at identifying which words form a semantic role. Lastly, we investigate if we can leverage the larger annotated dataset of Dutch laws by Bakker et al. (2022a) to identify the Flint roles in English law text. We hypothesise that having more data to fine-tune a model leads to better results on the test set. As such, we fine-tune a multilingual BERT model on Dutch data and test its ability to generalise to English.

1.3 Research questions and contributions

The main aim of this thesis is to automate part of the process of automatically extracting Flint frames from English law texts, specifically EU regulations. We direct our focus to labelling the Flint act roles, which are comparable to standard SRL roles, in sentences from English law text. Hence, the main question we aim to answer is:

> *How can we automatically label the roles of the Flint act frame in sentences from English legislative text?*

To support the process of answering our main research question, we will be looking to answer the following sub-questions in our research:

1. *Which methods already exist for semantic role labelling?*
2. *How do existing semantic role labelling roles relate to Flint act frame roles?*
3. *How can existing language models and techniques such as fine-tuning contribute to labelling Flint roles?*

4. *What effect does using a domain-specific or task-specific language model have?*
5. *Can we generalise a language model trained to label Dutch legal text to label English legal text?*

The experiments in this thesis are designed to answer these research questions. We expect that in the analysis of the experiments, we will find that by fine-tuning language models on a relatively small amount of annotated data, we can obtain credible results for labelling sentences with Flint roles. Moreover, we expect that fine-tuning language models pretrained on the legal domain or a span-detection task will provide even better results than using a general domain language model. Lastly, we believe that the multilingual BERT model generalises well enough to lead to competitive results when fine-tuned on a much larger Dutch dataset.

Apart from answering these research questions, we deliver a dataset with annotations of the Flint act roles on sentences containing an action, filtered from five EU regulations. This dataset is created through a thorough process of manual annotation, for which we provide a protocol containing a detailed set of guidelines. Both the dataset and the annotation protocol will potentially serve as a valuable resource for future research; the dataset can be used for further experimentation and the protocol can be used as a basis for similar annotation tasks in the legal domain. The dataset and the code for this thesis are available on GitLab[1].

1.4 Thesis outline

This thesis adheres to the following structure: chapter 2 provides an extensive overview of the relevant literature for this project. Chapter 3 details the methodology for our implementation of the different SRL models and in chapter 4 we present the results of these models. Lastly, we conclude in chapter 5 by answering the research questions, discussing the implications and limitations of our research, and identifying several directions for future work.

As a general remark, it should be noted that ChatGPT was used for support in generating the pgfplots in this thesis.

[1] https://gitlab.com/normativesystems/flintfillers/flintfiller-english

2 Related work

This chapter provides the theoretical background for the experiments in this thesis by giving an overview of the relevant literature. We will start this chapter by reviewing previous work on extracting information from legal text based on semantic annotations in section 2.1. Next, we will discuss several methods for formalising the norms from normative and legislative texts in section 2.2. In section 2.3 we give an overview of the development of language models over the past years and discuss the details of the language models that we will be using in the experiments in this thesis. Lastly, we will conclude this chapter with some background on semantic roles and the semantic role labelling task in section 2.4.

2.1 Information extraction from legal text

Over the years, we have seen different studies that use natural language processing techniques to extract and analyse information from legal text. In this thesis, we focus on semantic annotations to improve the modelling of legal text. This section highlights several studies that show how the field of extracting semantic information from law texts has developed from using mainly syntactic information to using semantic annotations for modelling text. Moreover, we d how transformers have recently been introduced as a tool to model legal text. At the end of the section, we will discuss the paper by Bakker et al. (2022a) which laid the foundations for the work in this thesis.

Early work in the field of legal information extraction was done by Uyttendaele et al. (1998) who presented the SALOMON project, which had the goal to automatically summarise legal texts to make them more accessible. The authors developed a methodology to extract relevant information from Belgian cases and use this information to construct the summaries for these cases. SALOMON used statistical techniques as well as semantic analysis to categorise cases and abstract the most relevant pieces of text from them.

Work on Italian law texts was done by Biagioli et al. (2005), who set out to explore a method that could make the process of retrieving information from legal documents more efficient. To this end, they developed a system, SALEM (Semantic Annotation for LEgal Management), that automatically enriches law texts with semantic annotations. This tool aims to classify paragraphs of law text based on their provision type (e.g. permission or obligation) and to extract specific strings of text that fulfil specific semantic roles within these paragraphs which can subsequently be assigned to a slot in the matching provision frame. The SALEM approach starts by syntactically preprocessing by POS-tagging and chunking the text. SALEM then uses several syntactic rules combined with semantic annotation rules to identify de-

pendencies within the paragraphs and semantically tag the text. On a set of 473 Italian law paragraphs, SALEM yielded promising results in the classification of provisions when compared to annotations that were done manually by law experts, with a precision and recall score of 97% and 96%, respectively. Moreover, on the task of semantic role extraction, SALEM scored 96% and 92% on precision and recall.

The paper by Brighi et al. (2008) adopts a similar rule-based approach in their attempt to fill slots in a semantic frame for modificatory provisions (e.g. replacements, deletions). They distinguish themselves from Biagioli et al. (2005) by using a deep rule-based parser, the Turin University Parser (TUP), to analyse the syntactic structure of modificatory provisions. This analysis of the syntactic structure is subsequently used by their semantic interpreter to fill slots in the semantic frame by using a set of pattern-matching rules.

In recent years, we have seen a shift from a strictly rule-based approach in automated legal information extraction to more advanced machine learning approaches. For example, the work by Gao and Singh (2014) implements a natural language processing and machine learning approach to automatically extract norms from contracts and subsequently specify them in a way that allows them to be used in multi-agent systems. Norms are formalised by assigning a norm type from the work of Singh (2014) to each norm and extracting the subject, object, antecedent, and consequent of the norm. The authors trained a logistic regression classifier on hand-annotated data to obtain the norm types, obtaining an F-measure of 84% on an independent dataset. A more detailed overview of the norm types used by Singh is provided in section 2.2.1. To extract the norm elements, they formulated four heuristics that rely upon the phrase chunks, POS tags, and dependency tags of the normative sentence. Table 2.1 shows an example of a norm and its elements extracted by Gao and Singh's method.

With the development of the Transformer architecture (see section 2.3.3) and transfer learning (see section 2.3.7), language models based on the Transformer architecture have been used increasingly for the task of modelling legal text. Shaghaghian et al. (2020) experimented with several language models based on BERT (Devlin et al., 2018) to perform document review tasks on legal texts. The experiments included tasks for information, fact, and rule navigation in legal texts and the authors found that for all these tasks, using a BERT-based model and tailoring it to the legal domain yielded the best results for navigating the information contained in law text.

A recent study by Bakker et al. (2022a) uses Flint frames to represent information extracted from Dutch legal text and proposes a rule-based as well as a transformer-based approach to fill these frames. Their work focuses on filling the institutional act frames. For brevity, we will refer to the institutional act frames as act frames throughout the rest of this thesis. For the transformer-based approach, annotated data on several Dutch laws are used to fine-tune BERTje (de Vries et al., 2019), a BERT (Devlin et al., 2018) model trained on the Dutch language. The model learns to tag data with labels that correspond directly to the slots to be filled in the Flint frames. Their rule-based approach uses a part-of-speech (POS) tagger to obtain the grammatical roles in the sentence and a chunk tagger to find the relations between words in the sentence. The resulting syntactic tags are used in combination with a set of rules to fill the Flint frames. The study shows promising

results for the transformer-based method as it reached an accuracy of 81% on the test set. The rule-based approach performed less well, reaching a 52% accuracy. Later work by Bakker et al. (2022b) improves upon their previous rule-based method by introducing rules that are based on POS tags as well as universal dependency tags (de Marneffe et al., 2021). The new model did better than the original rule-based model on recognising the Flint act roles, with an accuracy of 59%. In the next section, we will outline several approaches for formalising norms and we will provide a more elaborate overview of the Flint language.

2.2 Formalising norms

Several methods have been developed to formalise norms in order to create better models of normative texts. This section outlines some of these frameworks before narrowing down to the Flint language, the framework on which we will focus in the rest of this work.

2.2.1 Formalising normative relations

Early work in formalising normative relations has been done by Hohfeld (1917) who aimed to formalise norms to avoid the ambiguity between 'rights' and 'duties'. To do so, Hohfeld set out to formalise the relations between two adversarial parties in law cases and differentiated four legal concepts which exist exclusively in pairs: power-liability relations, immunity-disability relations, duty-claimright relations, and liberty-noclaim relations. These concepts form the basis of Hohfeld's framework to formalise the relations between two separate entities that both hold one of the rights. An example of a power-liability relation is when one individual has agreed with a second individual in a contract, that second individual has the power to enforce that contract. An immunity-disability relation might occur when a tenant has rent protection and the owner does not have the power to evict the tenant. Tax systems are examples of duty-claimright relations between citizens and government, where citizens must pay taxes and government has the right to claim taxes. Lastly, an individual's right to freedom of speech is an example of a liberty right, that cannot be taken away by another party. The work by Hohfeld can be used as a basis to conceptualise the legal relations between two parties as well as any other normative relation (van Doesburg, 2017).

Building on Hohfeld's work, Gao and Singh (2014) provide a framework for normative relationships that uses a fixed structure to represent a norm. Each norm includes a type, a subject, an object, an antecedent, and a consequent. This structure helps us understand the relationships and accountability between different agents or entities within a normative text. Singh's framework categorises normative relationships into six types: practical and dialectical commitments, authorisations, powers, prohibitions, and sanctions. Table 2.1 contains an example of an authorisation, according to Gao and Singh's method.

2.2.2 Frame-based approaches

Within Artificial Intelligence, the term *frame* was coined by Minsky (1974) as a "data structure for representing a stereotyped situation", which is now a funda-

Table 2.1: Formalisation of an authorisation (Gao & Singh, 2014)

Sentence	**"Plantronics shall have the right to audit such bill of materials and other information upon its request."**
Norm type	Authorisation
Subject	Plantronics
Object	null
Antecedent	its request
Consequent	Plantronics shall have the right to audit such bill of materials and other information

mental concept in knowledge representation. Formalising texts in frames helps us to get an overview of the events within the texts and their relationships with each other which makes it easier to identify the effects of actions. For example, the consequent of one event could 'trigger' the precondition of a different frame. Having norms formalised in such a system of frames, therefore, allows us to get a better understanding of the interactions within a piece of normative text. The work by Van Kralingen (1995) states that frames are an effective method for formalising interpretations for sources in natural language and proposes a frame-based approach for describing and modelling legal knowledge and normative relations. According to Van Kralingen, norms can be represented systematically by filling in text fragments from a source of norms describing their different dimensions and characteristics into slots in frames. The paper distinguishes three types of frames: norm-frames, act-frames, and concept-frames. Norm-frames are used to describe a norm and contain a slot that refers to a separate act-frame that describes the form of action. The concept-frames describe legal concepts, also known as institutional facts.

Another frame-based method was developed by Breaux (2009). The author's Frame-Based Requirements Analysis Method (FBRAM) produces a systematic overview of legal requirements that is meant to support knowledge engineers in building legal reasoning systems. The semantics of the legal requirements are specified in frames using a domain-independent upper ontology. The ontology contains sentence-level concepts such as permissions, obligations, and facts, as well as phrase-level concepts such as subject, object, and act.

One of the limitations of the Van Kralingen and FBRAM frames is that they do not effectively define the result of an act. Section 2.2.3 introduces Flint, a more recent frame-based method that addresses this issue.

2.2.3 Flint language

The Flint (Formal Language for the Interpretation of Normative Theories) language is proposed by Van Doesburg (2017) and van Doesburg and van Engers (2019) as a means to formally represent the interpretations of information extracted from normative text in frames. The original version of Flint distinguished three types of frames: institutional act frames, duty-claimright frames, and institutional fact frames. However, recent developments have led to a consolidation of the frames, where the duty frame is now integrated into the preconditions of the act frame. As a result, the Flint language currently distinguishes only the act and fact frame. The

components of these frames are shown in table 2.2.

Table 2.2: The Flint-language framework (van Doesburg, 2017)

Institutional act frame	**Institutional fact frame**
Act	Institutional facts
Actor	Derivation function
Object	
Recipient	
Precondition	
Creating postcondition	
Terminating postcondition	
Reference to source(s)	Reference to source(s)

Act frames are used to express a normative action performed by an actor upon an object, which might be to the benefit of some recipient. As such, the Flint act frame contains slots for the *action*, *actor*, *object*, and *recipient*. The *action* slot contains the action that is performed, it represents the thing that happens. The *actor* slot describes the agent that performs the action (with volition). The *object* slot contains the thing that is acted upon, in other words, that is undergoing the action and on which the action has an effect. The *recipient* slot shows who the intended target of an action is. Apart from these four roles, the frame also expresses the precondition(s) that must be satisfied to make the action legal and the result of the action, the postcondition, which must be different from the initial state. Table 2.3, taken from the paper by Van Drie et al. (2023), shows a version of an act frame of a sentence that does not have a recipient, with the pre- and postconditions excluded. In table 2.4 we provide an example of an act frame that does have a recipient.

Table 2.3: Act frame without the pre- and postconditions (van Drie et al., 2023)

Component	**Example**
Action	regulates
Actor	the Council
Object	the composition, powers and working methods of these committees
Recipient	-
Source text	The Council regulates the composition, powers and working methods of these committees and appoints the members.
Source	Art. 50 (3) of the Dutch health law

The institutional actions can result in the creation or termination of facts or, as per the original Flint framework, duties. Fact frames consist of descriptions of the preconditions of acts, where the state of the precondition is expressed by a function of facts. These functions can be Booleans or arithmetic functions describing the value of the precondition. In the outdated version of Flint, the duty frames contained a description of the duty itself, the duty holder, and the claimant. Moreover, a duty frame contained creating, enforcing, and terminating act frames. Creating act

Table 2.4: Act frame without the pre- and postconditions

Component	Example
Action	shall notify
Actor	Each Member State
Object	the provisions of its law which it adopts pursuant to paragraph 1 and any subsequent amendment affecting them
Recipient	the Commission
Source text	Each Member State shall notify to the Commission the provisions of its law which it adopts pursuant to paragraph 1, by 25 May 2018 and, without delay, any subsequent amendment affecting them.
Source	Art. 51 (4) of the GDPR

frames described the act that created the duty. Enforcing act frames expressed the acts a claimant could use to ensure fulfilment of the duty in case the duty holder refuses to satisfy the duty. Terminating act frames contained the acts that satisfy the duty, terminating the relationship between the duty holder and the claimant.

The paper by Van Doesburg and van Engers (2019) contains several examples of acts and facts from the Dutch Aliens Act for which the Flint framework is suitable. For example, an act described in the paper that could be contained in an act frame is 'granting a temporary regular residence permit'. This act is, among others, associated with the following fact: 'regular residence permit is granted from the day on which the alien has demonstrated that he meets all conditions'.

In this thesis, we will focus exclusively on the Flint language's act frames. An example of a manually filled-in act frame for an act in article 5 of the GDPR can be found in appendix A.1.

2.3 Language models

To automatically extract structured information from natural language sources such as law text requires a representation of textual data that allows it to be processed and analysed by machines. The problem of representing natural language is much more challenging than it would be for numerical data. For example, there is no straightforward approach to expressing how similar two words are, whereas the difference between two numerical values can easily be expressed in terms of another numerical value. Many different methods for capturing and expressing the information contained in natural language have been proposed, but in recent years we have seen the most success with Language Models (LM). Essentially, LMs are probabilistic models that predict how likely a certain word is to follow from a previous word or string of words. For example, when predicting the next word in the sequence *"I am swimming in the ..."*, a language model will likely assign a higher probability to the word *sea* than it will to the word *forest*. These models are useful tools for the application of many natural language tasks, including text generation, translation, summarization, and semantic role labelling. In section 2.3.7 we explain how basic language models can be adapted to perform these more complex downstream NLP tasks.

To represent the meaning of textual data, probabilistic LMs use continuous representations of words that can quantify the similarity between different words. Words are considered similar whenever the contexts they appear in are more similar, meaning they are surrounded by similar words. Bengio et al. (2003) first introduced word embeddings to represent this information in language models. Word embeddings are numerical vectors encoding the meaning of words in such a way that vectors that lie closer together in the vector space encode words that are more semantically similar. This representation allows us to extract the similarity as well as the relations between different words from the vector space. Well-known models that use word embeddings are Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and fastText (Bojanowski et al., 2017).

A limitation of the aforementioned models is that they learn a global word embedding for every word, yet the probabilities of words can vary greatly across different contexts. Because these probabilities are so context-dependent and the static word embeddings from Word2Vec or GloVe do not capture this information about context, language models could benefit from encoding information about the context of a token in the words embeddings, resulting in so-called contextualised word embeddings. ELMo (Peters et al., 2018) introduced deep contextualised word embeddings, which encode semantic and syntactic information of a token specific to its context. One of the most famous language models with contextualised word embeddings is BERT (Devlin et al., 2018). In this thesis, we will use BERT as well as adaptations of BERT for extracting the roles for the Flint act frames. In section 2.3.3 we give an overview of the workings of the Transformer architecture (Vaswani et al., 2017) on which BERT is based. We briefly touch on the transformer-based GPT models in section 2.3.4, before we provide a detailed description of BERT in section 2.3.5. Section 2.3.6 describes how BERT can be adapted for a specific domain.

2.3.1 Encoder-decoder models

Given that word order matters a lot in natural language, earlier language models looked at sequences one word at a time. Recurrent Neural Networks (RNN) were the state-of-the-art approach in language modelling for a long time. An RNN is a type of neural network (NN) architecture that for the prediction at the current time step, looks at the hidden states from all the previous time steps. This is suitable for language modelling because of the sequential nature of natural language. To predict the next word in a sequence, a recurrent model looks at all previous predictions in the sequence. Cho et al. (2014) introduced the RNN encoder-decoder, a neural network model consisting of two separate RNNs. The model has an encoder unit, which contains an RNN that takes the entire input sequence and reads each token in the sequence one by one, updating the hidden state of the RNN after each time step. At the end of the sequence, the model encodes the RNN's final hidden state into a fixed-length contextual vector representation, ignoring the intermediate hidden states. This vector encoding of the final hidden state can be considered as a summary of the input sequence and is subsequently fed to the decoder as input. The decoder unit has been trained to generate the output sequence one word at a time, conditioning on the words generated so far as well as the vector representation of the input sequence that was provided by the encoder.

Whereas the sequential nature of RNNs makes them suitable for processing nat-

ural language, it is also one of its biggest constraints. Dealing with longer sequences of text quickly becomes increasingly computationally expensive, so models are limited in the amount of text they can handle. This is because, during training of the RNN, the gradient of the error during backpropagation can become very small. The smaller this error of the gradient, the more difficult it becomes for the model to learn. This problem is known as the vanishing gradient problem and it makes it difficult for RNNs to handle long-term dependencies. As a result, in many cases the models will have already forgotten the beginning of the sequence by the time they reach the end of it, resulting in less accurate predictions. To address the vanishing gradient problem that RNNs encounter, Long Short-Term Memory (LSTM) models were developed (Hochreiter & Schmidhuber, 1997). LSTMs handle long-term dependencies because they contain cells that can store information over much longer periods of time and they use gates that decide which information they want to use to make predictions, which information they want to store, and which information gets discarded. Whereas LSTMs are able to deal with the issue that RNNs have with long-term dependencies, another problem is that the sequential nature of RNNs makes it impossible to parallelize them, making them slow to train and, consequentially, difficult to train on large amounts of data. In the next section (section 2.3.2), we discuss the attention mechanism, a parallel mechanism that also manages to deal with long-term dependencies in a more flexible way than LSTMs.

2.3.2 Attention

The idea of the attention mechanism was introduced by Bahdanau et al. (2014) with the aim to improve the performance of encoder-decoder models on the task of machine translation. The idea behind the attention mechanism is that when predicting the next word in a sequence, the model concentrates on the parts of the input sequence that contain the most relevant information. For example, we consider a model that wants to predict the next word in the sequence *"Every Thursday, Jan leaves his mother's house to have lunch at the office with his ..."*. Following the attention mechanism, a model would ignore the words *"his mother's"* and concentrate on the words *"the office"* as these contain the relevant information for predicting the next word in the sequence, which would likely be *"colleagues"*. To achieve this, the attention mechanism does not just rely on a summarization vector. Instead, the mechanism considers the relevance of all the words in the input sentence by taking a weighted sum of the hidden states of all the individual words to create the context vector. To compute this weighted sum of the hidden states, the mechanism first finds their attention scores, which determine the importance of these hidden states. The attention scores are calculated based on the current state of the decoder as well as the hidden states of the decoder. The attention scores are then normalised using a softmax function and subsequently, the weighted sum of the hidden states can be calculated. This process allows the decoder to shift its attention to the most relevant parts of the input sequence when predicting the target sequence. Figure 2.1 shows how the next target word y_t is predicted by conditioning on the predicted target words up to y_{t-1} and the weighted sum of the hidden states of the words in the input sequence $(x_1, x_2, ..., x_\top)$.

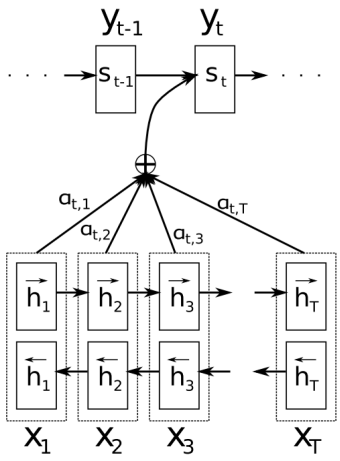

Figure 2.1: The Attention model (Bahdanau et al., 2014)

2.3.3 Transformers

To avoid the RNNs constraint of sequential computation Vaswani et al. (2017) proposed the Transformer, a new encoder-decoder model architecture that bypasses recurrence and relies exclusively on attention mechanisms. The Transformer, whose architecture is shown in figure 2.2, distinguishes itself from previous models in two ways: by using positional encodings and by using multi-headed self-attention. The positional encodings are included in the word embeddings to ensure that the model captures information about the word order in the source sequence since there is no recurrence. Self-attention is used as a means for finding dependencies within a certain sequence. Unlike the original attention mechanism that looks for dependencies between two sequences by shifting its attention to relevant parts of an input sequence to predict the next word of an output sequence, self-attention looks at other words in the same sequence to find the relations within that sequence when computing its context vector representation. This mechanism helps the Transformer with disambiguating words as it helps it to understand the context in which words can appear.

The Transformer's encoder stack is built up of six layers which each consist of a multi-head self-attention mechanism with a feed-forward neural network on top. The decoder stack is nearly identical to the encoder stack, but it contains an additional multi-head attention module on top of the feed-forward NN in each layer. The multi-head self-attention mechanisms in the model calculate the attention within a sequence several times in parallel and subsequently concatenate the outputs of these calculations. This allows the model to capture different kinds of relations within the same sequence. The Transformer has several advantages over encoder-decoder models based on RNNs. Since it is not constrained to sequential computation, the Transformer allows for parallelisation during training which leads to much faster training, which allows the model to train on much larger amounts of data. In section 2.3.4 and 2.3.5 we discuss the GPT and BERT language models, respectively, which are both based on the transformer architecture.

2.3.4 Generative Pretrained Transformers

The Transformer architecture was adopted by OpenAI to create GPT (Radford & Narasimhan, 2018), an open-source generative pre-trained language model. GPT had 117M parameters and was trained on the BooksCorpus dataset (Zhu et al., 2015) using a 12-layer decoder-only transformer. The BooksCorpus consists of 110380 books, which helped the model to learn long-term dependencies as books contain lots of contiguous text. GPT was compared to the GLUE multi-task benchmark (A. Wang et al., 2018) and achieved state-of-the-art results on 9 of the 12 tasks. The GPT language model was followed by the larger GPT-2 (Radford et al., 2019), which trained 48 layers on an even larger dataset of 40GB of text scraped from the Reddit platform and had 1.5B parameters, 10 times more than GPT-1. GPT-2 achieved state-of-the-art results on 7 out of 8 downstream tasks, showing that increasing the amount of training data and parameters improves the performance of the language models. Multiple versions of GPT-3 (Brown et al., 2020) were released in 2020. Its largest version has 175B parameters and 96 attention layers and was trained on the CommonCrawl (Zhang et al., 2020) dataset, containing 570GB of data. GPT-3 improved upon the previous models on machine translation, question-answering tasks, and generating news articles, but still struggles with natural language inference and text comprehension tasks. Notably, the first version of ChatGPT, OpenAI's chatbot application, was built on top of GPT-3.5. The most recent addition to the GPT series, GPT-4, is the largest GPT model to date and is the model behind the most recent versions of ChatGPT. GPT-4 yields better results than existing language models on benchmark tasks for language modelling and reaches human-level performance on exams (including a version of the Uniform Bar Examination) designed for humans. OpenAI did not disclose any information on the training process and model architecture such as the pretraining corpus, the number of parameters, and information on hardware, which they argue is due to the "competitive landscape and the safety implications of large-scale models like GPT-4" (OpenAI, 2023).

2.3.5 BERT

Devlin et al. (2018) introduced the language model BERT, short for Bidirectional Encoder Representations from Transformers. BERT's aim is to create contextual word embeddings, so the model only uses the encoder blocks from the Transformer architecture. To learn the context of words, BERT does not look at a sequence from left to right, right to left, or both, but uses a bidirectional encoder that performs self-attention in both directions. The authors introduced the Masked Language Modelling (MLM) training objective to allow for bidirectional training. The MLM approach randomly masks 15% of the words in the sequence with the [MASK] token and subsequently tries to predict the masked words by considering the other words in the sequence. Apart from the MLM task, BERT is also trained in next sentence prediction. During training, the model is shown different pairs of sentences and learns to predict whether the first sentence directly preceded the second sentence in the data. BERT uses WordPiece embeddings and a 30 000-token vocabulary. The model was pre-trained on a corpus consisting of the BooksCorpus (Zhu et al., 2015) which has 984M words as well as on the English Wikipedia, which has 2500M words. The paper by Devlin et al. (2018) reported two versions of the BERT model: BERT_{BASE} and BERT_{LARGE}. BERT_{BASE} has the same size as GPT, with

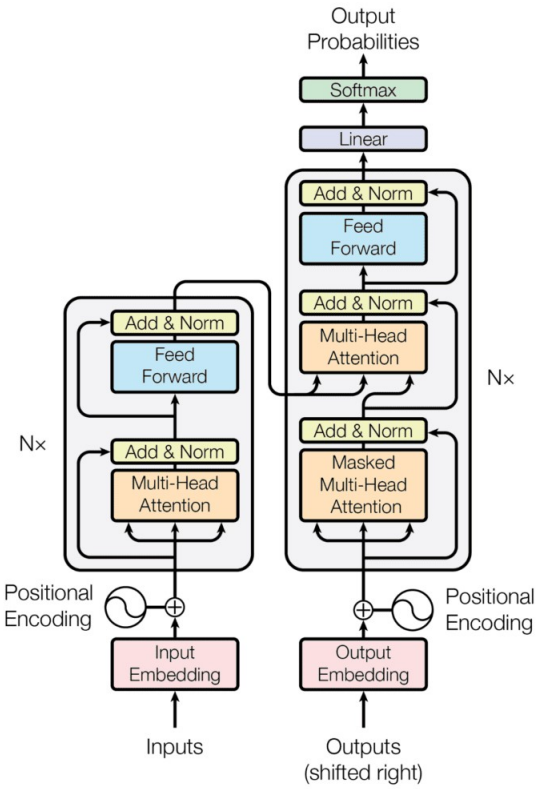

Figure 2.2: The Transformer architecture (Vaswani et al., 2017)

12 layers and 110M parameters. BERT_{LARGE} has 24 layers and 340M parameters. Both models were fine-tuned on 11 NLP tasks from the GLUE benchmark and both achieved state-of-the-art results on all tasks, with BERT_{LARGE} outperforming BERT_{BASE} in all categories. The fine-tuning of language models is discussed in section 2.3.7.

2.3.6 Alternative pretraining methods for BERT

Since the release of BERT in 2018, many modifications of BERT with variations on the original pretraining corpus and method have been published. This includes many models that have been tailored to perform well in a particular domain or at a specific task. A brief overview of these types of specialised BERT models, including the ones we used in the experiment for this thesis, is provided in this section.

2.3.6.1 Domain-specific language models

Using language models that have been pretrained on general-domain data alone may not result in the best performance on a domain-specific task. We can attribute this partly to the token distribution shift from corpora in the general domain to a specific target domain (Zheng et al., 2022), as well as the fact that there are many terms and phrases that have a different meaning in a domain-specific context than they would have in a general context (Shaghaghian et al., 2020). Moreover, domain-specific texts, especially legal texts, often have a different syntactic structure than general domain texts. Efforts to deal with these semantic and syntactic differences

between general-domain and in-domain texts have shown that creating domain-specific language models can significantly increase performance on downstream NLP tasks in the relevant domain (Beltagy et al., 2019; K. Huang et al., 2019; Lee et al., 2019; Yang et al., 2020; Zheng et al., 2022).

There are two common approaches in pretraining domain-specific language models: the continual pretraining of a general-domain pretrained language model by using pretrained initial weights from the general domain, or starting with random initial weights and pretraining the model from scratch on in-domain data. Domain-specific language models are typically built using the first approach, starting out with a language model pretrained on general-domain data after which the model is further pretrained on in-domain data. BioBERT (Lee et al., 2019), a language model for biomedical corpora, adopts this approach. BioBERT is initialised with the weights from BERT, after which it is pretrained on PubMed abstracts (4.5B words) and PMC articles (13.5B words). BioBERT was fine-tuned on named entity recognition (NER), relation extraction (RE), and question answering (QA) tasks in the biomedical domain and its performance on these tasks was compared to BERT. The results show that BioBERT significantly outperforms BERT on all tasks, illustrating the advantage of using an in-domain corpus.

Within the architecture, engineering, and construction (AEC) domain, Zheng et al. (2022) present RegulatoryBERT and CivilBERT, two BERT-based models further pretrained on in-domain and close-domain corpora, respectively. They compare the performance of these models to that of BERT on text classification (TC) and NER tasks. The results show that RegulatoryBERT performs better than BERT in all experiments and achieves state-of-the-art results on the TC and NER tasks in the target domain. CivilBERT performs worse than RegulatoryBERT on all tasks and sometimes even performs worse than BERT, showing that further pretraining on close-domain data is not sufficient for improving performance on these NLP tasks in the AEC domain.

Gu et al. (2021) challenge the assumption that out-of-domain text is still useful for pretraining domain-specific language models. The authors argue that, given that there is enough text of the target application domain available, language models pretrained from scratch on domain-specific text can be superior to language models pretrained on mixed-domain data on downstream NLP tasks in the target domain. They pretrain a BERT model from scratch on 14 million PubMed abstracts, resulting in PubMedBERT. A comparison of the performance of PubMedBERT to that of BioBERT on several biomedical NLP tasks such as question answering, named entity recognition, relation extracting, and document classification shows that PubMedBERT makes gains in performance over most NLP tasks.

SciBERT (Beltagy et al., 2019) uses the BERT architecture but is pretrained from scratch on 1.14M papers from Semantic Scholar from the computer science domain (18%) and the biomedical domain (82%). SciBERT outperforms BERT-base on tasks in the biomedical and computer science domain, as well as on multidomain tasks. Moreover, SciBERT achieved state-of-the-art results on several tasks in the biomedical domain and matches or outperforms BioBERT on all evaluated tasks.

Work in the legal domain has been done by Chalkidis and Kampas (2019), who released Law2Vec, a collection of two models with word embeddings of 100 and 200 dimensions. Law2Vec was based on Word2Vec's skip-gram model and was trained using 5-word windows on 123066 documents with UK, EU, US, Canadian, and Aus-

tralian legislation and court decisions, resulting in models with a total vocabulary of 169439 words. Research based on the BERT language model in the legal domain has been done by Shaghaghian et al. (2020). The authors use a general domain token set, a legal token set, and a hybrid token set to pretrain BERT models, both by using the original BERT weights and by using random initial weights. Their results show that models that were customised to the legal domain had superior performance on passage retrieval, text similarity, and sentiment analysis tasks. Moreover, they showed that using Legal Tokens improves performance when training from scratch compared to using General Tokens, but that further pretraining on the original weights results in the best performance. Elwany et al. (2019) show that using a BERT model fine-tuned on large amounts of legal data improves performance compared to a simple neural network.

The largest language model for the legal domain, on which we will focus in this thesis, is LEGAL-BERT (Chalkidis et al., 2020). The authors released a collection of BERT models customised on the legal domain for downstream legal NLP tasks. They used a dataset of 12GB of English legal data from different legal domains, including contracts, legislation, and court cases. An overview of the models that have been released can be found in table 2.5. LEGAL-BERT-FP uses the weights of BERT_{BASE} to further pretrain on legal corpora for up to 500k steps. LEGAL-BERT-SC uses the BERT_{BASE} architecture and is trained solely on legal corpora. LEGAL-BERT-SMALL is a smaller version of LEGAL-BERT-SC with 6 layers, 512 hidden units, and 8 attention heads. The authors use the EURLEX57K, ECHR-CASES, and CONTRACTS-NER datasets to evaluate the models on text classification and sequence tagging tasks. LEGAL-BERT-FP and LEGAL-BERT-SC have comparable results on all three datasets, both consistently outperforming the BERT_{BASE} model. The LEGAL-BERT-SMALL performs only slightly worse than the bigger LEGAL-BERT models, functioning as a competitive alternative. Apart from these models that were released by Chalkidis et al. (2020), three sub-domain versions of LEGAL-BERT were released by AUEB's Natural Language Processing Groups on huggingface[1]. These models were pretrained on sub-corpora of the original pretraining corpus with the idea that they perform better on tasks in their respective domains.

Table 2.5: Overview of the six versions of LEGAL-BERT that have been released thus far, including their training corpora and their source.

Model	Training corpora	Source
LEGAL-BERT-SC	Full 12GB set	Chalkidis et al. (2020)
LEGAL-BERT-FP	Full 12GB set	Chalkidis et al. (2020)
LEGAL-BERT-SMALL	Full 12GB set	Chalkidis et al. (2020)
CONTRACTS-BERT-BASE	US contracts	AUEB's NLP group
EURLEX-BERT-BASE	EU legislation	AUEB's NLP group
ECHR-BERT-BASE	ECHR cases	AUEB's NLP group

[1]https://huggingface.co/nlpaueb/legal-bert-base-uncased

2.3.6.2 Additional pretraining objectives to learn structural information

Apart from the BERT models that have been designed to perform well in a specific domain, work has been done to pretrain versions of BERT that better capture certain structures and dependencies in text.

For example, StrucBERT (W. Wang et al., 2019) takes the underlying structures that are present in natural language into account during pretraining to improve performance on tasks such as question answering and sentiment classification. StructBERT considers language structure both within and between sentences by including two additional pretraining objectives to encode the dependencies between words and between sentences. To find structures on the word level, StructBERT shuffles words during pretraining to train the model to place them into their correct order. Moreover, the original BERT model's next sentence prediction pretraining task is extended by making the model predict not only the next but also the previous sentence, thereby getting a better sense of sentence structures within natural language. StructBERT achieved 93% F1 on the SQuAD 1.1 question answering task and reached a state-of-the-art 89% average score on the GLUE benchmark.

Recently, SpanBERT (Joshi et al., 2020) was introduced to improve performance on predicting spans of text, which would be suitable for NER, question answering, and coreference resolution tasks. The pretraining process of SpanBERT differs from that of BERT in several ways. First of all, instead of randomly masking 15% of the tokens in the pretraining text, SpanBERT masks random spans of tokens in the text, thereby also masking 15% of the tokens in total. Secondly, the model is trained to predict spans based on the token representations of the tokens at the boundaries of the spans, ignoring individual representations of tokens within the masked span. Lastly, SpanBERT does not incorporate next-sentence prediction as one of its pretraining objectives. The original version of BERT was pretrained by looking at two sequences and determining whether the two are related. However, the authors of SpanBERT believed that the model will be able to understand the text better if it is presented with single, longer sequences that provide more context. Moreover, they believe that presenting the model with two potentially unrelated sequences from different contexts will only open the model up to more noise. As such, SpanBERT uses single-sequence training with sequences of up to 512 tokens. SpanBERT was tested on a variety of tasks and made significant gains compared to BERT on question answering and coreference resolution tasks. SpanBERT obtained F1 scores of 94.6% and 88.7% on SQuAD 1.1 and SQuAD 2.0, respectively. Additionally, SpanBERT set a new state-of-the-art on the CoNLL-2012 coreference resolution task with an F1 of 79.6%.

2.3.6.3 Multilingual models

The authors of BERT also released Multilingual BERT (M-BERT) (Devlin et al., 2018), a language model pretrained on multiple languages. The training corpus for this model consisted of the combination of the Wikipedia content of the 104 most common languages and the model utilises a single, multilingual word piece vocabulary. One of the main advantages of M-BERT is that it allows for cross-lingual transfer learning: models can be fine-tuned on task-specific annotated data in a certain language, to be evaluated on that task with test data in a different language (see section 2.3.7 for a more detailed description of transfer learning and

fine-tuning). The extent of this cross-lingual generalisation quality was tested by Pires et al. (2019), who designed several fine-tuning experiments to investigate M-BERT's capabilities to perform this transfer across languages. For the experiments, they used the CoNLL-2002 and -2003 sets for NER and the CoNLL 2017 task for POS. M-BERT obtained an F1 score of over 59% on all language pairs in the CoNLL sets and achieved an accuracy of over 80% for the pairings between four languages (English, German, Spanish, and Italian) on the POS experiment. The authors looked into how much the performance of the model is dependent on the lexical overlap between languages, i.e. on overlapping word pieces between the fine-tuning and the evaluation language. Moreover, they investigated how the model performed for languages that are written in different scripts, where there is zero lexical overlap. They found that M-BERT's performance is comparable for many different levels of overlap and that the model even performs well for languages with different scripts, suggesting that its ability to generalise across languages is not simply a result of the overlapping vocabularies of languages, but can be attributed to its learned capacity to capture and represent a wide range of linguistic patterns and structures.

2.3.7 Fine-tuning word embeddings

Language models can be used to perform a wide variety of downstream NLP tasks. In the past, a common approach was to train an entire language model on a specific task. A common problem with this approach is that it requires large amounts of annotated data to train the model for the NLP task, which is not always available. Moreover, the learned model often does not generalise well to other domains. To circumvent the constraint of limited amounts of data, fine-tuning pretrained word embeddings is commonly used as a technique where we take a model that has already been trained and then use a small set of new data to continue training this model for the intended task. Fine-tuning is a technique that is derived from transfer learning, an idea that was introduced in the field of machine learning by Bozinovski and Fulgosi (1976) for training neural networks. The core idea of transfer learning is that we can leverage the knowledge that is gained by a model trained to perform one task and use it to create a model to solve a related task, where we use the same model with a different dataset to learn the new task. For example, a model that has been trained to recognise images of chairs can be used to train a model to recognise images of tables.

Fine-tuning has proven to be an effective method for training language models for downstream NLP tasks by using pretrained word embeddings and fine-tuning for a specific task by using a small set of (annotated) data. Labutov and Lipson (2013) developed a method to re-embed words for a supervised task. They used source embeddings and a small set of annotated data to learn embeddings specific to a sentiment classification task for movie reviews. Their results showed that when finetuning embeddings from a hierarchical log-bilinear model (Mnih & Hinton, 2008) as well as embeddings from a neural model (E. Huang et al., 2012), the performance on the task of classifying movies as good or bad improves upon the performance of their baseline models that re-embedded a set of zero-vectors and a set of uniformly distributed random vectors. This is especially the case when fine-tuning on training sets that contain less than 5000 examples. The work by Gao and Ichise (2017) uses two deep neural networks to reduce the dimensionality of Word2Vec vectors

and improve the cosine similarity of synonyms, using a supervised as well as an unsupervised approach. The new, adjusted embeddings improved over the original Word2Vec embeddings in terms of cosine similarity of synonyms and also reduced the distance between non-synonymous word pairs.

Howard and Ruder (2018) proposed a new method for fine-tuning language models called Universal Language Model Fine-tuning (ULMFiT). The ULMFiT approach consists of three steps. The first step is concerned with pretraining a language model on the general domain. The second step fine-tunes the language model to the target task domain. The authors use a discriminative fine-tuning approach for this step, where different layers are fine-tuned with different learning rates, dependent on the type of information the layer contains. The third step fine-tunes the model for the target classification task. The authors experimented with the ULMFiT approach on six different datasets on sentiment analysis, question classification, and topic classification tasks. The results showed that models trained with the ULMFiT method did not overfit and reached state-of-the-art results on all six tasks, even for datasets with as little as a 100 training examples.

Shi and Lin (2019) fine-tuned pretrained BERT models for relation extraction (RE) and semantic role labelling (SRL) that does not rely on any syntactic features. They used a pretrained BERT model with an LSTM and a Multi-Layer Perceptron (MLP) on top for fine-tuning. For both tasks, the LSTM was used to learn contextual information and the MLP was used to make predictions for the concerned classification task. The method by Shi and Lin achieved a precision of 73.3% on the TAC Relation Extraction Dataset. For the SRL task, the authors' method achieved state-of-the-art F1 scores on the CoNLL 2005 in-domain and out-of-domain datasets of 88.8% and 82%, respectively. On the CoNLL 2012 benchmark, the model achieved competitive results but was not able to match the results by Ouchi et al. (2018), who achieved an F1-score of 87%. The model by Ouchi et al. achieved these results by fine-tuning ELMo (Peters et al., 2018) word embeddings as well as the network's weights.

In this thesis, we are interested in fine-tuning our models on a semantic role labelling task. An overview of this task is provided in section 2.4.

2.4 Semantic role labelling

We introduce the semantic role labelling task on which we will focus in this thesis. Section 2.4.1 introduces semantic roles and puts forward several approaches to automated semantic role labelling, concluding with the transformer-based approach on which we will be relying in our experiment. In section 2.4.4, we consider the elements of Flint act frames and how they map to well-known semantic roles.

2.4.1 Semantic roles

Semantic role labelling (SRL) is the task that is concerned with finding the meaning of a sentence by determining "who did what to whom" by understanding the relations within a sentence (Jurafsky & Martin, 2021). SRL helps to understand the semantics of the different constituents of a sentence by labelling them with semantic roles. Typically, the SRL process consists of two steps: identifying the predicate of the sentence and identifying the predicate arguments and their function or role in

the sentence. The predicate of the sentence is the main verb in the sentence that indicates the action that is performed. After establishing the predicate of a sentence, SRL aims to identify the different arguments in a sentence and their relationship to the predicate (Palmer et al., 2005). These roles that arguments have in relation to the predicate of a sentence are the semantic roles. Semantic roles, as we are familiar with them today, were first introduced in the 1960s by Gruber (1965) and Fillmore (1968), in the form of thematic roles. Thematic roles are general, human-readable semantic roles that capture the abstract relationship of the argument to the predicate without delving into the specific meaning of the arguments. For example, in the sentence '*Bob threw the ball*', *threw* is the predicate, *Bob* is the agent (the volitional causer of the event) and *the ball* is the theme (the entity affected by the event). As of right now, there does not exist a definitive list of semantic roles, but some of the most common thematic roles are provided in table 2.6.

Table 2.6: Common thematic roles (Jurafsky & Martin, 2021)

Thematic Role	**Definition**
AGENT	The volitional causer of an event
EXPERIENCER	The experiencer of an event
FORCE	The non-volitional causer of the event
THEME	The participant most directly affected by an event
RESULT	The end product of an event
CONTENT	The proposition or content of a propositional event
INSTRUMENT	An instrument used in an event
BENEFICIARY	The beneficiary of an event
SOURCE	The origin of the object of a transfer event
GOAL	The destination of an object of a transfer event

2.4.2 Semantic role labelling resources

Several verbal resources have been developed that provide standardised annotation schemes and large labelled datasets for the semantic role labelling task. The first was FrameNet (Baker et al., 1998), a large English lexical database that was based on the theory of frame semantics by Fillmore (1968). The idea behind FrameNet is that it captures semantic concepts (e.g. events, relations, entities) in frames and that each frame has a specific set of semantic roles associated to it, the frame elements (FEs). These frames can be invoked by certain words, which are the frame's lexical units. For example, the frame *being_employed* has as frame elements an *employee*, an *employer*, a *field*, a *place of employment*, a *position*, and a *task*. The frame has several lexical units by which it can be invoked, such as *work* and *temp* (verbs) or *job* (noun). This example demonstrates how FrameNet not only considers verbs but also other types of words (e.g. nouns or adjectives) that capture the same semantic concepts, represented in the frames. FrameNet describes over 1200 semantic frames and their associated semantic roles. Moreover, it contains over 200 000 sentences that have been assigned to one of the frames and annotated with the corresponding semantic roles.

One of the inconveniences of FrameNet is that it assigns a unique set of semantic roles to each frame, resulting in thousands of roles. This makes the framework complex, tougher to train and learn for a model, and hard to generalise to other languages. A resource that does not encounter these limitations as it uses only six core semantic roles is The Proposition Bank (PropBank) (Palmer et al., 2005). The PropBank is an SRL resource with semantic labels on all of the sentences from the Penn TreeBank. It does not consider nouns and adjectives like FrameNet, but assigns a set of senses to each verb, and associates its set of semantic roles to each verb sense. These roles have a specific function for each individual verb sense and are numbered as `Arg0`, `Arg1`, `Arg2`, etc. In general, the `Arg0` and `Arg1` roles correspond to the thematic agent and patient, respectively, but exceptions do occur. To illustrate how these semantic roles might relate to a predicate, an example for the verb *increase*, taken from Jurafsky and Martin (2021), is included below:

increase.01 "go up incrementally"
`Arg0`: causer of increase
`Arg1`: thing increasing
`Arg2`: amount increased by
`Arg3`: start point
`Arg4`: end point

PropBank's main limitation is that its roles do not always indicate the same type of relation to the predicate. For example, the verbs *eat* and *feel* both have an `Arg0`, but for *eat* this argument expresses the *agent* whereas for *feel* it expresses the *experiencer* (Di Fabio et al., 2019).

VerbNet (Schuler & Palmer, 2005) implements thematic roles to indicate the relationship between the arguments and the predicate of sentences. This way, it sidesteps FrameNet's limitation of having a large number of highly specific roles, as well as PropBank's limitation of having roles that can express multiple types of relationships within sentences. However, one of the main drawbacks of VerbNet is its limited coverage of only 6791 verbs, which makes it hard to generalise to other domains as domain-specific verbs are likely to be missing.

Most recently, VerbAtlas was introduced by Di Fabio et al. (2019) as a semantic resource that, similar to FrameNet, structures semantic concepts in frames. However, rather than assigning roles that are unique to each frame, it uses thematic roles for the frame elements which allows for better generalisation across frames. The VerbAtlas frames consist of groups of verbs that express similar semantic concepts, for which the resource uses WordNet synsets as a basis. WordNet (Fellbaum, 1998) is a lexical database that has structured words into synsets, which are groups of words that have meanings that are similar. VerbAtlas groups synsets that are closely related to create its frames, resulting in 466 frames.

2.4.3 Semantic role labelling approaches

Given that knowing the semantic roles within a sentence can help us better understand the relations within and between sentences, there has been a growing interest in automatically labelling words in the sentence with their semantic role. Over the years, many different approaches to the SRL task have been developed, ranging from

rule-based to machine learning and hybrid approaches.

A rule-based approach was developed by Gildea and Jurafsky (2002). The authors trained statistical classifiers on over 50 000 FrameNet (Baker et al., 1998) sentences. Their method used syntactic parsing and handwritten rules to assign semantic roles to the constituents of the sentences, achieving an accuracy of 82%. One of the main limitations of rule-based systems is that they are only as good as their rules. Incomplete or low-quality rules can significantly decrease the performance of a rule-based system. Several machine learning approaches from the early 2000s focused on Support Vector Machines (SVM). For example, Mitsumori et al. (2005) used an SVM with additional semantic-class features to solve the CoNLL-2005 shared task and resulting in an F1 score of 71%.

Later attempts to improve the performance of SRL algorithms focused on neural approaches, adopting an end-to-end approach that directly extracts the semantic roles from the text. Using a BiLSTM architecture resulted in state-of-the-art results on the CoNLL 2005 and CoNLL 2012 sets (He et al., 2017). Most recently, pretrained transformer-based language models such as BERT have been used for the SRL task by fine-tuning them on annotated data. As was touched upon in section 2.3.7, Shi and Lin (2019), reached state-of-the-art results on the CoNLL 2005 test set.

2.4.4 Flint act frame roles

The first four components of the Flint act frame - the *action*, *actor*, *object*, and *recipient* - bear a close resemblance to the more traditional thematic semantic roles. Hence, Van Drie et al. (2023) compares the process of assigning these four Flint roles to spans in sentences to the task of semantic role labelling, where the focus is specifically on labelling sentences that contain an act with the Flint roles instead of labelling any sentence with traditional semantic roles. They state that the Flint roles can be assigned to action sentences by considering the predicate-argument structure of the sentence, equating the predicate to the *action*, and assigning the other three roles to the arguments of that predicate. In their comparison of Flint roles to thematic roles, they relate the *actor* to the thematic *agent*, the *object* to the *patient* and *theme* thematic roles, and the *recipient* to the *beneficiary*. As such, the task is to determine: who *(actor)* did *(action)* what *(object)* to whom *(recipient)*? An overview of this mapping is contained in table 2.7.

Table 2.7: Overview of the Flint components linked to thematic roles (van Drie et al., 2023)

Flint component	Thematic role	Definition
Actor	Agent	The initiator of some action, capable of acting with volition
Object	Patient	The entity undergoing the effect of the action
Object	Theme	The entity which is moved by the action, or whose location is described
Recipient	Beneficiary	The entity for whose benefit the action was performed

In this thesis, we leverage and build upon the research discussed in this chapter. Specifically, as a baseline, we implement a rule-based method based on the work by Bakker et al. (2022b) that was presented in section 2.1. Moreover, we fine-tune BERT (discussed in section 2.3.5) and several domain-specific and task-specific models (discussed in section 2.3.6).

3 Methods

For this thesis, our aim was to explore various methods and language models for extracting the semantic roles that can be used to build Flint act frames from an action sentence. We limited our focus to classifying the *action*, *actor*, *object*, and *recipient* roles, a decision that was motivated by the fact that these four elements often fall within the confines of a single sentence and form the core of the Flint act frame. We can consider this classification task a semantic role labelling task given that the Flint act frame roles are very similar to the semantic roles commonly used in different SRL tasks. Section 2.4.4 provided more insight into the semantic roles associated with the Flint act frame, their functions, and their relationship to the more widely known thematic semantic roles.

We hypothesised that fine-tuning a transformer-based language model on the Flint labels of interest would yield better results than using a standard SRL approach and mapping the resulting labels to Flint. Moreover, we expected that using language models that were pretrained on legal text or that were pretrained to predict spans of text would lead to even better results on our legal semantic role labelling task. Therefore, in our exploration, we experimented with mapping the labels of an out-of-the-box BERT-based SRL model to Flint labels as well as fine-tuning several variations of the BERT language model on labelling action sentences with the aforementioned Flint roles. Additionally, we fine-tuned a multilingual BERT model on a much larger set of Dutch data, to see if the increased amount of data would lead to better results on our English test set. The performance of all of these models was compared to a rule-based baseline model, which we based on the Dutch rule-based model by Bakker et al. (2022b).

The process of collecting, preprocessing, and annotating the data for our experiment is described in section 3.1. The experimental setup is discussed in section 3.2 and section 3.3 outlines the details of the methods of evaluation that we used to analyse the annotator agreement and the performance of our models.

3.1 Dataset

As was discussed in section 2.3.7, fine-tuning a language model on a specific task required a (small) set of data with annotations specific to the target task. Therefore, we required a dataset with action sentences from several EU regulations, where each sentence in the dataset was annotated with the four different Flint semantic roles that we were interested in. Sections 3.1.1, 3.1.2 and 3.1.3 dive into the methods for assembling this dataset, motivating and justifying the decisions that were made in the data collection and data annotation process. In section 3.1.4, we discuss the details of the dataset for Dutch law texts (van Drie et al., 2023), which we used to

fine-tune Multilingual BERT.

3.1.1 Collection of the data

The sentences for this dataset were collected from five different EU regulations. The regulations that were used for this dataset are the General Data Protection Regulation (GDPR), the Digital Services Act, the Digital Markets Act, the Capital Requirements Act, and the Food Safety Act. We collected the XML files of these law texts, with the exception of the GDPR, from the *EUR-Lex* website[1], the official resource for European Union law. We parsed the XML files for these regulations to obtain the raw text and split this into sentences, leaving us with 5857 sentences. To collect the sentences from the GDPR, we used the dataset with CLAL (Core Legal Annotation Language) annotations on the GDPR from Nazarenko et al. (2022), which is publicly available on the *Northern Paris Computer Science Lab* website[2]. Their dataset consists of annotations in the CLAL language on the entire GDPR, compiled in XML format. We chose to use this dataset with CLAL annotations because its specific format was easier to parse and preprocess. This was especially the case for extracting sentences that are contained in enumerations. In the CLAL dataset, enumerations are clearly marked as such, which made it easier to identify their elements and to extract sentences from them. An example of an annotation of a sentence from the GDPR in the CLAL language from this dataset can be found in listing 1. There are no datasets with CLAL annotations for any other EU regulations available, which is why we collected the remaining data from the *EUR-Lex* website. Parsing the CLAL XML file resulted in 807 sentences from the GDPR, yielding a total dataset of 6664 sentences from EU regulations.

For this thesis, we were specifically interested in sentences from law texts that contain an action. Therefore, after collecting all the separate sentences from the EU regulations that we selected, we manually selected the sentences that contain an action. During this filtering, we looked for actions that would specifically influence or shape the behaviour of individuals or organisations. In the legal context, this is known as a normative effect and this concept was recommended as a selection criterion by Van Drie et al. (2023). For example, we consider 'he grants the application' a normative action, but 'he is aware of the decision' is not a normative action. After filtering, we were left with a final dataset of 1575 action sentences, which was 23.6% of the total collected sentences. The distribution of the number of sentences over the various regulations in the final dataset and the versions of the regulations that were used can be found in table 3.1.

While we cannot be certain that we selected all sentences containing an action from the original five EU regulations, the dataset contained a reasonable amount of sentences for fine-tuning a language model for our intended task. Moreover, the aim of this experiment was not to detect every single action sentence but rather to gain insight into the extent to which a model can recognise Flint roles in an action sentence. As such, we considered this dataset to be sufficient to complete the task and to draw meaningful conclusions after annotating and fine-tuning.

[1] https://eur-lex.europa.eu/homepage.html
[2] https://lipn.univ-paris13.fr/ fl/CLAL/V2-2022-09/Light_GDPR_EN.xml

```
<PARAG IDENTIFIER="002.002">
<NO.PARAG>2.</NO.PARAG>
<leg:COMPLEMENT IDENTIFIER="002.002.001" type="impact"
 ↪  rel="LEGAL_TEXT">
<P>This Regulation does not apply to the processing of personal
 ↪  data:</P>
<LIST TYPE="alpha">
<ITEM> (a) in the course of an activity which falls outside the scope
 ↪  of Union law; </ITEM>
<ITEM> (b) by the Member States when carrying out activities which
 ↪  fall within the scope of Chapter 2 of Title V of the TEU; </ITEM>
<ITEM> (c) by a natural person in the course of a purely personal or
 ↪  household activity; </ITEM>
<ITEM> (d) by competent authorities for the purposes of the
 ↪  prevention, investigation, detection or prosecution of criminal
 ↪  offences or the execution of criminal penalties, including the
 ↪  safeguarding against and the prevention of threats to public
 ↪  security. </ITEM>
</LIST>
</leg:COMPLEMENT>
</PARAG>
```

Listing 1: Annotations in the Core Legal Annotation Language on a paragraph of Article 2 of the GDPR. The annotation makes it easier to extract the separate items of an enumeration and to append each one individually to its header.

3.1.2 Annotation of the data

To fine-tune the language models in this experiment for the task of recognising the Flint semantic roles in action sentences, we required the dataset of action sentences to be enriched with annotations of these Flint roles on the token level. We asked five annotators to identify four Flint act-frame roles in the sentences in our dataset: *action*, *actor*, *object*, and *recipient*. It is important to note that these roles may span multiple tokens and that not every sentence may contain all four of the roles. During the annotation process, annotators were presented with action sentences from the dataset and were asked to indicate which words they considered to be part of which of the four aforementioned roles. Any words that they did not indicate as being part of one of these roles were automatically labelled as *other*.

Annotation setup

The annotations were performed with the help of the TNO annotation tool, which provided a user-friendly interface for selecting the words that correspond to a particular role (see figure 3.1). This tool was developed for the study of extracting Flint roles from Dutch law texts by Bakker et al. (2022a). It was built using the Vue[3]

[3]https://vuejs.org/

Table 3.1: Overview of the regulations used, the versions consulted and the number of action sentences in the final dataset.

Regulation	Version	Sentences
Capital Requirements Regulation (EU) No. 575/2013	01-01-2023	437
Digital Markets Act (EU) 2022/1925	14-09-2022	251
Digital Services Act (EU) 2022/2065	19-10-2022	279
Food Safety Regulation (EC) No. 178/2002	01-07-2022	152
General Data Protection Regulation (EU) 2016/679	04-05-2016	456

and Vuetify[4] frameworks and it utilises the RecogitoJS[5] library for the annotations (van Drie et al., 2023). After signing into the annotation tool, the annotators were presented with an extensive annotation protocol with instructions and examples to ensure consistent annotations across all annotators, which was based on the instructions for the experiment by Bakker et al. (2022a). The protocol detailed what is meant by the different semantic roles and included a list of instructions on which types of words and constructions to include in and exclude from the annotations. The full annotation protocol can be found in appendix A.2.

Before starting the real annotations, annotators received 8 practice sentences that contained examples of issues covered in the protocol to familiarise themselves with the annotation tool and to ensure that they understood the instructions correctly. While annotating, annotators were able to return to the annotation protocol at any time. Moreover, annotators were able to save their annotations and sign in and out of the tool at their own accord.

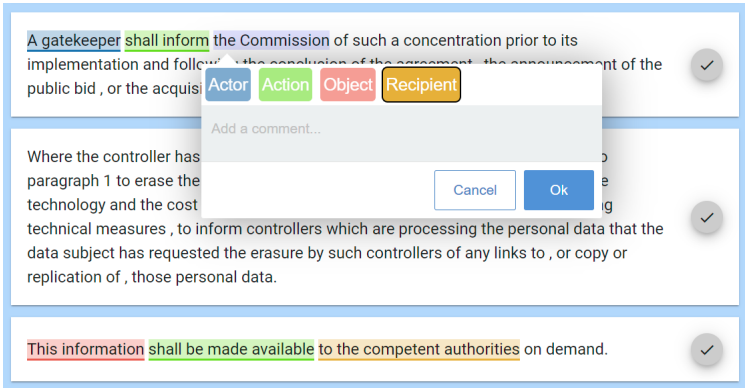

Figure 3.1: The annotation tool interface. Annotators can select words and assign them to one of the labels. They are also given the option to leave a comment, justifying their decision.

The annotations for this experiment were carried out by the author of this thesis and four other interns in the TNO Data Science department, all of whom were master students. This group consisted of four Dutch native speakers, one Bulgarian native speaker, and no English native speakers. Moreover, none of the annotators had any legal background. Before starting the annotation process, we randomly selected 200 sentences from the dataset that were to be annotated by all five annotators to

[4]https://vuetifyjs.com/en/
[5]https://github.com/recogito/recogito-js

determine the inter-annotator agreement. The author annotated an additional 775 unique sentences, and all six other annotators annotated an additional 150 unique sentences.

Annotator agreement

After completing the annotation process, we were left with a set of 2335 annotated sentences. We computed the inter-annotator agreement for the 196 unique sentences that were annotated by all five annotators on the token level. We used Fleiss' kappa metric (Fleiss, 1971) to determine the inter-annotator agreement. Fleiss' kappa represents the agreement between multiple raters when assigning items to classes or categories and is defined as

$$\kappa = \frac{\overline{P} - \overline{P}_e}{1 - \overline{P}_e} \tag{3.1}$$

where $\overline{P}$ is the observed agreement among annotators, $\overline{P}_e$ is the chance-corrected agreement, $1 - \overline{P}_e$ reflects the degree of agreements that can be expected above chance and $\overline{P} - \overline{P}_e$ reflects the degree of agreement that was really achieved above chance.

In addition to computing the inter-annotator agreement over the entire dataset, we were interested in observing the agreement per specific Flint role. This would allow us to gain insight into where annotators might have had more trouble understanding the annotation protocol or which aspects of the annotation task might have proved to be more ambiguous. To compute the agreement per category, we implemented the specific agreement coefficient, which was first introduced by Dice (1945) as a means to determine the relationship between two different animal species and later recognised and refined by Uebersax (1982) to function as a metric for determining inter-rater reliability per category for two or more raters. This index reflects for each category the probability that a randomly chosen rater assigns an item to a certain category, given that another randomly chosen rater also assigns that item to that category. In our specific case, it represents for each Flint role, the probability that a randomly chosen annotator assigns a token in a sentence to a specific Flint role, given that another randomly chosen annotator also assigned that token to that same Flint role. The specific agreement for a Flint role is computed as

$$SA_k = \frac{\sum_{i=1}^{n'} r_{ik}(r_{ik} - 1)}{\sum_{i-1}^{n'} r_{ik}(r_i - 1)} \tag{3.2}$$

where n' reflects the number of tokens that were annotated by two or more annotators, r_{ik} is the number of annotators that assigned token i to Flint role k, and r_i is the number of annotators that assigned token i to any Flint role.

3.1.3 Preprocessing of the annotated data

After completing the annotation process, we collected all annotations for further preprocessing. For the sentences that were annotated by multiple annotators, we kept only one annotation. In case all annotations by the different annotators were

the same for a single sentence, we kept that annotation in our dataset once. For the sentences where there was disagreement among annotators, we randomly selected which annotation to keep, making sure there was an even distribution across annotators. After this selection process, we were left with a single annotation for 1539 sentences, which is 97.7% of the 1575 sentences that were set aside for annotation and 66% of the 2335 annotations that we had at the end of the annotation process. To determine if there was any noteworthy class imbalance in our annotated dataset, we computed the number of tokens per label for the annotated set as well as the number of annotated roles. The results are reported in section 4.1.2.

Preprocessing for model fine-tuning

To create the datasets to fine-tune all the variations of the BERT language model that we selected, we used each model's respective tokeniser to tokenise the annotated sentences. After tokenisation, the tokens and tags were misaligned since BERT tokenisers add the [CLS] and [SEP] tokens to the beginning and end of the sequence and split some words up into multiple tokens. We realigned the tokens and the tags by adding the *other* tag for the [CLS] and [SEP] tokens and by adding extra instances of the tag for words that were split up by the tokeniser. Subsequently, we encoded all labels as integers and we split the dataset for training, validation, and testing, using a 80-10-10% split. This split allowed us to make the most efficient use of our relatively small dataset, as it provided a reasonable number of training data while leaving a sufficient quantity for reliable and meaningful test results. All BERT models were trained and validated on the same set of sentences. Moreover, for our final evaluation, we used the same 10% split of sentences to test our models, including the rule-based and role mapping model (sections 3.2.1 and 3.2.2, respectively).

3.1.4 Dutch dataset

Based on the promising results of Multilingual BERT on NLP tasks that were discussed in section 2.3.5, as well as the fact that increasing the amount of training data has a positive effect on model performance, we hypothesised in section 1.2 that fine-tuning a multilingual BERT model on a larger set of Dutch annotated data could lead to better results when testing on English data. To test this hypothesis, we fine-tuned the Multilingual BERT model on the Dutch dataset that was presented by Van Drie et al. (2023). This dataset consists of 4463 annotated sentences from 55 Dutch laws. The sentences were annotated with the same four Flint roles by four annotators with a linguistic background, based on the annotation protocol on which we based the protocol for our English annotations. The inter-annotator agreement for this dataset was $\kappa = 0.75$, which falls within the substantial category (Landis & Koch, 1977).

3.2 Experimental setup

For this work, we implemented and compared several models for the task of extracting Flint roles from action sentences, utilising a diverse set of methodologies and techniques. This section provides an extensive overview of the models that we used. In section 3.2.1, we introduce our implementation of the rule-based model

from Bakker et al. (2022b), which relies on linguistic patterns to assign Flint roles to spans of text within a sentence. We used this model as our baseline to provide insight into the complexity of the task and to use it as a reference point against which the performance of the transformer-based models could be measured. We continue by discussing our approach for a model, hereafter referred to as the **mapping model**, that harnesses an out-of-the-box SRL model and maps its roles to Flint roles in section 3.2.2. Finally, we investigated the effect of using the annotated data described in section 3.1 to fine-tune five variations of the BERT language model on the task. The setup for these models is discussed in section 3.2.3.

We conclude this section with an overview of all of the implemented models in table 3.4.

3.2.1 Rule-based baseline model

As a baseline, we implemented a rule-based model that leverages the syntactic structure of the sentence to identify Flint roles, based on the method by Bakker et al. (2022b). Our methodology consisted of tagging our sentences with POS- and dependency tags and subsequently applying rules based on these tags to assign the relevant Flint roles. To tag the sentences, we used the part-of-speech-tagger and the dependency tagger from the `en_core_web_sm` model from the spaCy[6] library. Subsequently, following the original paper's methodology, we used the POS tags to create chunks. These chunks formed different phrases, allowing a role to be assigned to multiple words, rather than singular tokens. We then used the rules formulated in the original paper to assign Flint roles to the sentences. The roles are applied per sentence and in order and are formulated as follows:

1. If a token in a phrase carries the *nsubj* or *obl:agent* dependency tag, the entire phrase will be assigned the *actor* role.

2. If a token in a phrase carries the *dobj* or *nsubjpass* dependency tag, the entire phrase will be assigned the *object* role.

3. If a token in a phrase carries the *dative* dependency tag, the entire phrase will be assigned the *recipient* role.

4. If a token in a phrase carries the *root*, *ccomp* or *xcomp* dependency tag, the entire phrase will be assigned the *action* role.

All tokens that are not assigned a Flint role by one of these rules are automatically labelled as *other*. Table 3.2 contains a simplified overview of how the dependency tags are related to the Flint roles for the rules used for our baseline model.

3.2.2 Role mapping model

Because of the state-of-the-art results achieved by Shi and Lin (2019) on SRL tasks using BERT (section 2.4) and because of the analogy of thematic roles and Flint roles by Van Drie et al. (2023), we wanted to explore how we could leverage an existing SRL implementation based on a BERT model to label action sentences with Flint roles. The key idea behind this model is to use a model that has been

[6]https://spacy.io/

Table 3.2: The table shows how the dependency tags are related to the Flint roles according to Bakker et al. (2022b). The rules for the baseline models consider whether a phrase in the sentence contains one of these dependency tags and assigns the associated Flint role to the phrase.

Dependency tag	**Flint role**
nsubj *obl:agent*	Actor
dobj *nsubjpass*	Object
dative	Recipient
root *ccomp* *xcomp*	Action

pretrained on a standard SRL task to label the sentences in our dataset and to map the resulting semantic roles to Flint roles. To implement this model, we required two things: an out-of-the-box BERT SRL model and a mapping from that model's semantic roles to Flint roles.

For this experiment, we used a version of the language model BERT that had already been fine-tuned on an SRL task. We used the pretrained model from the `transformer-srl`[7] library by Riccardo Orlando from the Sapienza University of Rome. This model performs the semantic role labelling task based on the method by Shi and Lin (2019), using PropBank semantic roles. The model is the BERT$_{BASE}$ language model, fine-tuned using the CoNLL 2012 dataset (Pradhan et al., 2012), where it achieved an F1 score of 86% on the test set. We used this model to return for each sentence in our dataset the predicate sense and the accompanying predicate-argument structure for each possible predicate in the sentence. For sentences with multiple potential predicates, we needed to determine what the predicate was in order to apply our subsequent mapping to the correct version of the initial model's labelling. To identify the predicate of each sentence, we used the dependency parser component from the spaCy `en_core_web_sm`[8] model to determine the root of each sentence, which we took to be the predicate. From the output by the `transformer-srl` model, we then selected the labelling with this predicate as the correct labelling. For sentences where the root of the sentence, as indicated by the dependency parsing component, was not one of the potential predicates found by our SRL model, we labelled each token in the sentence as *other* and used this as the final labelling. For the other sentences, we continued to use the PropBank labelling for the predicate-argument structure of our selected predicate to apply our subsequent mapping.

As was discussed in section 2.4.4, there is a relatively straightforward mapping from thematic semantic roles to Flint roles. However, there is no straightforward way to directly map PropBank roles to Flint roles. Therefore, we required a method to translate the PropBank output from our SRL model to thematic roles before we

[7]https://github.com/Riccorl/transformer-srl
[8]https://spacy.io/models/en

could map to the intended Flint roles. To achieve this, we used the mapping provided by the VerbAtlas resource, which contains a mapping from PropBank semantic roles to the corresponding VerbAtlas semantic roles for 5306 PropBank predicate senses. We applied this to all sentences whose determined predicate sense had a mapping in the VerbAtlas resource. For the sentences for which there was no mapping in the resource for that sentence's predicate sense, we labelled each token in the sentence as *Other* and use this as the final labelling.

After obtaining the thematic roles for each sentence, we used another mapping to translate from thematic roles to our final Flint roles. To create this mapping, we used the overview of how thematic roles can be linked to Flint by Van Drie et al. (2023), discussed in table 2.7 in section 2.4.4, as a basis. After going over all the VerbAtlas thematic roles, we added three more mapping pairs to the overview by Van Drie et al. (2023):

- Topic $\mapsto$ Object
- Asset $\mapsto$ Object
- Recipient $\mapsto$ Recipient

The *topic* semantic role is defined in the VerbNet documentation as a 'theme characterised by information content transferred to another participant (specific to events of communication)'. Given that the *topic* is another form of a *theme*, which is mapped to the *object* Flint role by Van Drie et al. (2023), we map the thematic *topic* role to *object* in Flint. Considering VerbNet's definition of the *asset* role as a 'value that is a concrete object', we also map the thematic *asset* role to *object* in Flint. Finally, we mapped the thematic *recipient* to *recipient* in Flint. VerbNet defines the *recipient* as a 'destination that is animate', often seen with verbs of change of possession or verbs of communication.

The complete mapping from thematic roles to Flint roles that we adopted can be found in table 3.3. Given that this is an n-to-1 mapping, where there are no thematic roles that map to more than one Flint role, this did not cause any issues in our translation of the roles.

Table 3.3: Final mapping from VerbAtlas thematic roles to Flint roles

VerbAtlas thematic role	**Flint role**
Agent	Actor
Patient Theme Topic Asset	Object
Beneficiary Recipient	Recipient

For our final mapping model implementation, we created a pipeline that takes a set of sentences as input, labels each sentence with the `transformer-srl` model, maps the resulting PropBank roles to thematic roles and subsequently to Flint roles, and finally returns each sentence in the set together with a list containing the Flint role per token.

3.2.3 Fine-tuning models

As was discussed in section 2.3.7, fine-tuning pretrained language models, more specifically BERT, on specific tasks using relatively small sets of annotated data has proven to be an effective method for various downstream NLP tasks. Moreover, studies using variations of the BERT model that were pretrained for a specific domain (e.g. legal) or task (e.g. predicting spans) also proved to be very effective in their respective domain (Chalkidis et al., 2020; Joshi et al., 2020). As such, we selected the following variations of the BERT model to fine-tune on the Flint role labelling task:

- **BERT** Captures contextual information from the left and right context.
- **LEGAL-BERT** Pretrained on data from the legal domain. Potential to better understand patterns in our legal dataset.
- **EURLEX-LEGAL-BERT** Pretrained on data from EU regulations. Potential to better understand patterns in our legal dataset with EU regulations.
- **SpanBERT** Pretrained to capture the relationships between spans of words. Potential to better performance if fine-tuned on an SRL task.
- **Multilingual BERT** Integrates cross-lingual transfer learning. Potential to leverage larger Dutch dataset.

The first four models were fine-tuned on the training and validation set of our English dataset (section 3.1.3). We fine-tuned Multilingual BERT, hereafter referred to as **M-BERT**, on the larger Dutch dataset that was discussed in section 3.1.4, to evaluate its ability to generalise to English for our labelling task.

To monitor the performance of our models during fine-tuning, we used the cross-entropy loss function. This function captures how far apart the true and the predicted probability distributions across the classes are and feeds this information to the model in order to minimise the difference between these probabilities, thereby minimising the errors made by the model. The cross-entropy loss is defined as

$$L_{CE} = -\sum_{i=}^{n} t_i \log(p_i) \tag{3.3}$$

where n is the number of classes, t_i is the truth label and p_i is the probability of the i^{th} class.

Because the classes in the Dutch dataset by Bakker et al. (2022a) showed a certain degree of imbalance across the different Flint roles and because our dataset was annotated on similar texts and based on similar annotation instructions, we also expected to see a class imbalance in our dataset. To avoid a bias toward the majority class(es), we implemented class weighting by giving a set of weights to the loss function that ensured that the loss function shifts more focus to minimising the error for the minority classes. We balanced the weights by assigning to each role label a weight that is inversely proportional to the frequency of that role label in the train and validation sets. As such, we calculated the weights for each class as

$$W_k = \frac{n_{samples}}{n_{classes} \times n_k} \tag{3.4}$$

where $n_{samples}$ represents the total number of tokens in the set, $n_{classes}$ represents the number of classes (Flint roles) and n_k is the number of tokens with the role label for which we are trying to determine the weight.

We fine-tuned each model for 4 epochs as this is the recommended number of epochs for BERT model fine-tuning (Devlin et al., 2018). To find the best settings for the learning rate and batch size, we executed a grid search to systematically explore the best combination of values. For the values of these hyperparameters, we followed the recommendations of the authors of BERT, who suggested the learning rate values of 5e-5, 3e-5, and 2e-5 and batch sizes of 8, 16, and 32 for model fine-tuning. Because the fine-tuning process took approximately 45 minutes per model, we were able to perform an exhaustive search across all value combinations for all models. For the final fine-tuning of each model, we used the combination of hyperparameter settings with the highest accuracy on the validation set after the fourth epoch. We used the same train-validation-test split to fine-tune each of the models.

Table 3.4: Overview of the implemented models

Model	**Description**
Rule-based (baseline)	Using a set of heuristics based on POS tags and dependency tags to assign Flint roles
Mapping model	Mapping the PropBank roles from an out-of-the-box SRL model to thematic roles to Flint roles
BERT fine-tune	Fine-tuning the BERT language model on annotated data
LEGAL-BERT fine-tune	Fine-tuning a BERT language model pretrained on legal data on annotated data
EURLEX-LEGAL-BERT fine-tune	Fine-tuning a BERT language model pretrained on EU legislation on annotated data
SpanBERT fine-tune	Fine-tuning a BERT language model that was pretrained to better predict spans of text on annotated data
Multilingual BERT fine-tune	Fine-tuning a multilingual BERT model on a larger set of Dutch annotated data to test on English sentences

3.3 Evaluation

We employed several metrics to evaluate the performance of our models, all of which provide insight into a different aspect of the models' performance. To evaluate the models' ability to classify the Flint roles on the token level, we calculated the accuracy as well as the balanced accuracy. The accuracy represents the ratio of the

Table 3.5: The different scenarios that might occur when comparing the predicted output of a system to the annotations.

Scenario	Explanation
I	Surface string and role type match
II	System hypothesises a role
III	System misses a role
IV	System assigns wrong role type
V	System gets boundaries of surface string wrong
VI	System gets boundaries and role type wrong

correctly predicted words to the total number of words in the dataset. However, accuracy is not a suitable metric for datasets with class imbalances, we will report the balanced accuracy metric, which is the average of the accuracy per Flint role, which gives a fairer representation of the models' performance. Moreover, we used precision, recall, the F1 scores per class, the macro F1 score, and the weighted F1 score. The F1 scores per class allow us to assess how well the models can identify specific roles. The macro F1 score averages this performance across all classes and the weighted average allows us to take class imbalance into account. Hence, this weighted average allowed us to get a more realistic sense of the overall model performance as we expected quite some imbalance in the classes. For example, the *action* role typically takes up fewer tokens than the *object* role, which can consist of multiple phrases. Moreover, we expected class frequency to vary because a large portion of the data would be likely to be classified as *other*, given that in many cases large parts of the sentence (e.g. pre- and post-conditions) do not fall under any of the four relevant Flint roles.

Considering the ultimate goal of filling Flint frames with complete roles extracted from legal text, we wanted to go beyond our evaluation of classifying roles on a token-tag-based schema and assess how well our models performed at recognising roles as a whole within sentences. To this end, we also chose to incorporate in our analysis the metrics used for the SemEval-2013 Task 9.1 (Segura-Bedmar et al., 2013) based on the error types from the Fifth Message Understanding Conference (MUC-5) evaluation (Chinchor & Sundheim, 1993). These metrics, typically used for Named Entity Recognition (NER) tasks, do not consider performance on the token level, but rather on the role level. They provide insight into whether all tokens belonging to a certain role were recognised by a system and whether the system assigned the correct role type, taking partial matches into account. The key idea behind these metrics is that when we compare the predictions by a system to a golden standard (our annotations) we can encounter six different scenarios for how well the system performed. The scenarios are described in table 3.5. Scenario I describes a situation where a model correctly captures the boundaries and the role type of a role that it identified in comparison to the golden annotation. Scenario II represents a situation where a system predicted a role but where there is no role in the golden annotation and scenario III is when a system completely misses a role that is present in the annotation. Scenario IV happens when a system finds a complete role but does not assign the right type to that role, and scenario V occurs when a system finds a role but gets the boundaries wrong (it assigns too many or

too few tokens to the role). The last scenario, VI, happens when a system does not get the boundaries and the role type of an identified role right.

To express these scenarios in evaluation metrics, MUC-5 distinguishes five different error types for comparing model predictions to annotations: correct (COR), partial (PAR), incorrect (INC), missing (MIS), and spurious (SPU). The definitions for these error types are provided in table 3.6. Whereas these error types closely resemble the scenarios described in table 3.5, these metrics cannot be used directly to measure the performance, as we are interested in these errors for both the string matching and the role type assignment simultaneously. For example, it remains unclear whether to assign the correct, partial, or incorrect error type if the system gets the boundaries of a role completely right, but assigns the wrong role type. To deal

Table 3.6: The error types for the MUC-5 evaluation (Chinchor & Sundheim, 1993).

Type	**Definition**
Correct	System output and annotation are equivalent
Partial	System output and annotation are similar but not equivalent
Incorrect	System output and annotation do not match
Spurious	System output produces response that does not exist in annotation
Missing	Annotation is not captured in system output

with this limitation, the SemEval-2013 defined four evaluation schemas for boundary and type matching to measure the precision, recall, and F1 score of a system based on the MUC-5 error types, where the error types are assigned differently for each schema. Table 3.7 contains an overview of the SemEval-2013 evaluation schemas and table 3.8 shows how the error types interact with the different scenarios and schemas. For all evaluation schemas, an exact match of boundaries and role type (scenario I) is classified as correct, a missing role (scenario II) is classified as spurious, and a hypothesised role (scenario III) is classified as missing. For the other three scenarios, we observe some differences in what is considered correct or incorrect. For example, if a model gets the boundaries of a role correct but it assigns an incorrect role type, the *partial* and *exact* schemas still consider the prediction correct, whereas the *type* and *strict* schemas consider it to be incorrect. Whenever a model assigns the correct type to a role but not the correct boundaries (scenario V), the *type* schema considers the prediction correct, the *partial* schema considers it partially correct, and the *exact* and *strict* schemas (which always require an exact boundary match) consider it incorrect. Lastly, in case a model only detects are role but with the wrong boundaries and the wrong role type, it is partially correct according to the *partial* schema. All other schemas consider the prediction to be incorrect.

The counts of the error types are used to calculate the performance of the model. However, to compute the precision, recall, and F1 score for the different evaluation schema, two more quantities are defined: the number of annotations in the test set and the number of outputs produced by the system under evaluation. The first is

[9] https://pypi.org/project/nervaluate/

Table 3.7: The evaluation schemas defined by the SemEval-2013 (Segura-Bedmar et al., 2013).

Schema	Explanation
Strict	Exact match on boundary of surface string and role type
Exact	Exact match on boundary of surface string, regardless of role type
Partial	Partial match on boundary of surface string, regardless of role type
Type	Some overlap between system output and annotation is required

Table 3.8: Overview of how the MUC errors, evaluation schemas and possible scenarios interact. Based on an example from the nervaluate documentation page[9].

	Annotation		Prediction					
Scen.	**Role**	**String**	**Role**	**String**	**Type**	**Partial**	**Exact**	**Strict**
I	Actor	the EBA	Actor	the EBA	COR	COR	COR	COR
II			Actor	the EBA	SPU	SPU	SPU	SPU
III	Actor	the EBA			MIS	MIS	MIS	MIS
IV	Actor	the EBA	Object	the EBA	INC	COR	COR	INC
V	Actor	the EBA	Actor	EBA	COR	PAR	INC	INC
VI	Actor	the EBA	Object	EBA	INC	PAR	INC	INC

defined as:

$$\text{POSSIBLE}(POS) = COR + INC + PAR + MIS = TP + FN \tag{3.5}$$

The latter is defined as:

$$\text{ACTUAL}(ACT) = COR + INC + PAR + SPU = TP + FP \tag{3.6}$$

Subsequently, the precision, recall, and F1 can be computed based on the possible and actual number of annotations. How the precision and recall metrics are defined depends on the evaluation schema that is applied. In scenarios where we require an exact match (strict and exact), we compute:

$$\text{Precision} = \frac{COR}{ACT} = \frac{TP}{TP + FP} \tag{3.7}$$

$$\text{Recall} = \frac{COR}{POS} = \frac{TP}{TP + FN} \tag{3.8}$$

In scenarios where we are looking for a partial match (partial and type), precision and recall are calculated as:

$$\text{Precision} = \frac{COR + 0.5 \times PAR}{ACT} \tag{3.9}$$

$$\text{Recall} = \frac{COR + 0.5 \times PAR}{POS} \tag{3.10}$$

For our analysis of our models' performance, we computed the overall scores for each evaluation schema. We also computed the scores for each evaluation schema per role type to get a better idea of the roles our models may have struggled with, and the type of error they tended to make per role.

4 Results

In this chapter, we report the results of our exploration for a system that labels legal text with the semantic roles associated with the Flint act frame, according to the methodology outlined in the previous chapter. We start in section 4.1 by presenting the dataset that we obtained after the annotation process and reporting the inter-annotator agreement. Then, in section 4.2, we discuss the fine-tuning process of the various BERT models. Finally, we evaluate the performance of all of the implemented models in section 4.3.

4.1 Dataset

In order to fine-tune the five variations of the BERT language model that we selected and to evaluate the performance of all seven of the models that we implemented, we required annotations on a set of action sentences from European Union regulations. During the annotation process, five annotators each annotated a portion of the 1575 sentences that we selected. The author annotated 975 sentences and the other four annotators were all given 350 sentences. Not all annotators completed all annotations, resulting in a dataset of 2335 annotated sentences. The inter-annotator agreement on this dataset is presented in section 4.1.1. After performing duplicate removal on the set of annotated sentences to ensure that only one annotation per sentence was used in the fine-tuning and evaluation process, we were left with a dataset of 1573 sentences with annotations. The distribution of the roles in this dataset is reported in section 4.1.2.

4.1.1 Annotator agreement

To determine the reliability of our annotations, we computed the inter-annotator agreement using the Fleiss' kappa metric. This metric represents the level of agreement between annotators over what would be expected by chance. Once the annotation process was completed, 196 sentences were annotated by all five annotators. For these sentences, the annotator agreement was $\kappa = 0.712$, which is substantial agreement (Landis & Koch, 1977). This result shows a general consistency and reliability across the annotations, suggesting that the annotators understood the task and the annotation guidelines correctly.

To get a more nuanced evaluation of the annotator agreement and to understand where most disagreements occurred in the annotation process, we considered the inter-annotator agreement per category using the specific agreement metric. The specific agreement reflects — for each Flint role — the probability that a randomly chosen annotator assigns a token in a sentence to a specific Flint role, given that

another randomly chosen annotator also assigned that token to that same Flint role. The results of the specific agreement for each category are shown in table 4.1.

Table 4.1: The specific agreement per label for the annotated dataset. The metric represents the probability of a randomly chosen annotator to assign a token in a sentence to a certain category given that another randomly chosen annotator also assigned that token to the same category.

	Action	**Actor**	**Object**	**Recipient**
Specific agreement	0.938	0.946	0.713	0.708

For the specific agreement for the roles in our dataset, we observe relatively high agreement for all Flint roles, especially for the *action* and *actor* roles, with scores of 0.938 and 0.946, respectively. Whereas the agreement for the *object* and *recipient* roles is also satisfactory, it is substantially lower than the others, suggesting that the annotators had a more difficult time identifying the *object* and *recipient*.

Actor - Action - Object - Recipient

Annotator A:
[PRECONDITION], the Commission may consult third parties.

Annotator A:
[PRECONDITION], the Commission may consult third parties.

Figure 4.1: An example sentence from the Digital Markets Act, with annotations by two annotators. For brevity, we have omitted the precondition from the sentence.

Figure 4.1 shows an example where annotators did not agree on the role of the argument 'third parties' in relation to the action 'consult'. It indicates that the annotators had different interpretations of how these 'third parties' function in relation to the verb 'consult', showing they may have had trouble understanding what constitutes an *object* or a *recipient* in this case. Another type of disagreement is shown in figure 4.2, where the annotators did not agree on the boundaries for the *object*. The long and complex nature of this role may have caused the annotators to have different understandings of what should be included.

These results suggest that to improve the agreement for these specific roles, clearer instructions and examples in the annotation protocol may be required or that the guidelines for these roles may need to be redefined altogether. However, we should also consider that these roles may simply be more challenging to recognise than the *actor* and the *action*.

We note that the lower agreement on the *object* and *recipient* roles might also be exacerbated in the results of the models, as the inconsistencies may make it more challenging for the fine-tuned models to learn and generalise.

Actor - Action - Object - Recipient

Annotator A:
To carry out inspections, the Commission may request the assistance of auditors or experts appointed by the Commission pursuant to Article 26(2), as well as the assistance of the national competent authority of the Member State, enforcing the rules referred to in Article 1(6) in whose territory the inspection is to be conducted.

Annotator B:
To carry out inspections, the Commission may request the assistance of auditors or experts appointed by the Commission pursuant to Article 26(2), as well as the assistance of the national competent authority of the Member State, enforcing the rules referred to in Article 1(6) in whose territory the inspection is to be conducted.

Figure 4.2: An example sentence from the Digital Markets Act, with annotations by two annotators.

4.1.2 Distribution of roles in the dataset

To gain insight into the class distribution in our final dataset and to identify potential class imbalances, we present the number of annotated per role type and the total number of roles per role type in table 4.2. For a visual insight into the class distribution in the dataset, we plotted the bar graphs in figure 4.3.

Table 4.2: The number of tokens that were annotated per role type and the number of total roles that were annotated per role type. Annotators did not actively label tokens as *other*; they were automatically labeled as such if they were not annotated. There is no data for the *other* label for the number of roles as it is not an actual role type.

	Action	**Actor**	**Object**	**Recipient**	**Other**
Number of tokens	4016	5519	17277	3064	31810
Number of roles	2061	1390	1551	481	-

The left plot in figure 4.3 displays the number of tokens in the dataset that were annotated as each role. The right plot shows how often a certain Flint role was identified by the annotators. Both plots show a certain class imbalance, which may have implications for the performance of the models and for the way we can evaluate their performance.

In the plot on the left, we see that the *other* category has the most tokens assigned to it. This category consists of the tokens that were not selected by the

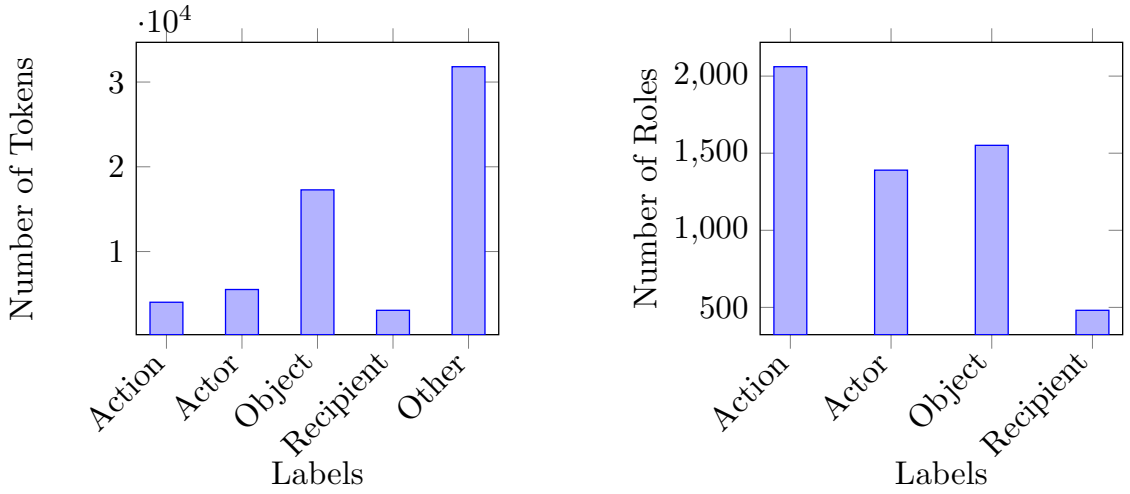

Figure 4.3: Distribution of tokens and roles across Flint labels in annotations. The plot on the left shows the number of tokens that were annotated with each label in the final dataset. Note that the *other* label is automatically assigned to tokens that are not annotated as one of the Flint roles. The plot on the right shows for each Flint role how many of them were annotated in total in the final dataset.

annotators as belonging to any Flint role and were thus automatically labelled as *other*. This shows that most tokens in the sentences in our dataset did not belong to any of the four Flint act frame roles that we were looking for. We also observe that the *object* category has a much larger number of tokens annotated than the *actor*, *action*, and *recipient* categories, which could indicate one of two things: either there are many more *object* roles present in the dataset, or *object* roles typically consist of more tokens.

In the plot on the right, which displays how often each individual type of role has been assigned in the total dataset, we also observe an imbalance in the classes. We notice that there is approximately the same number of *actor* and *object* roles as sentences in our dataset, suggesting that the sentences in our set typically contain one *actor* and one *object*. When looking into the numbers of tokens per role, we found that for the *actor* and the *action*, 84% and 99% consisted of 1-5 tokens. For the *object*, 42% of the roles lay in the 1-5 range, with 58% spanning more than 5 tokens. This also shows that out of the two scenarios described above, the idea that *object* roles are typically comprised of more tokens is the correct one. The action role occurs a little over 2000 times in our set, which shows that sentences can contain more than one action. Lastly, we observe that the *recipient* role occurs far less than the other three roles, just under 500 times. This indicates that not all sentences contain a *recipient* or that during the annotation process, annotators were less focused on identifying the *recipient* in sentences. The performance of our fine-tuned models can potentially be influenced by this relative scarcity of *recipient* roles as the models will have fewer training examples to learn the patterns of this specific role. To limit the influence of these class imbalances, we implemented class weights in the loss function in the process of fine-tuning our models, as was discussed in section 3.2.3.

4.2 Fine-tuning of the BERT models

After we obtained our annotated dataset, we used it to fine-tune the four BERT models that we selected that were only pretrained on English natural language[1] that we selected on the Flint role labelling task. Moreover, we used the dataset with annotations on Dutch legal text from Van Drie et al. (2023) to fine-tune the Multilingual BERT model. In this section, we discuss the results of the hyperparameter optimisation for each of the models and we present their fine-tuning losses.

4.2.1 Hyperparameter optimisation

To maximise the performance of the fine-tuned models, we performed a grid search to find the best settings for the hyperparameters. In our search, we focused on finding the best balance between the learning rate and the batch size, for which we selected ranges of values that were recommended by the BERT authors (Devlin et al., 2018). As such, nine combinations of settings were tested for each model. We used an 80-10-10% split on our dataset for training, validation, and testing, and we evaluated the results on the validation accuracy after fine-tuning for four epochs.

The results of the grid search for BERT and LEGAL-BERT are presented in figures 4.4 and 4.5, respectively. In the figures, the plot on the left contains the validation accuracies plotted against the learning rates, and the plot on the right shows the relation between the different bath sizes and the validation accuracy. Each point in one of the plots corresponds to the point with the same validation accuracy in the other plot, which can be recognised by the colour and shape of their mark. In the figures, the combination of settings with the highest validation accuracy is indicated with a red circle. As such, we observe that the best combination of hyperparameters for fine-tuning the BERT model is a learning rate of 3e-5 and a batch size of 8, and for LEGAL-BERT a combination of a 5e-5 learning rate and a batch size of 8 is best. The complete overview of the results of the hyperparameter optimisation sweeps for all fine-tuned models is included in appendix A.3.

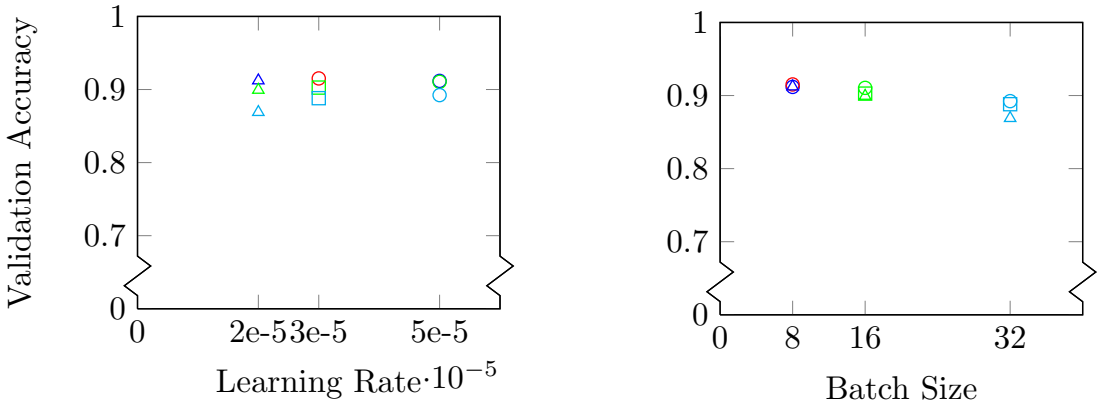

Figure 4.4: BERT hyperparameter optimisation results

In the results of the grid searches for the BERT models that were fine-tuned on the English annotated dataset, we noticed a trend that the validation accuracy

[1]We fine-tuned BERT, LEGAL-BERT, EURLEX-LEGAL-BERT, and SpanBERT on our English dataset.

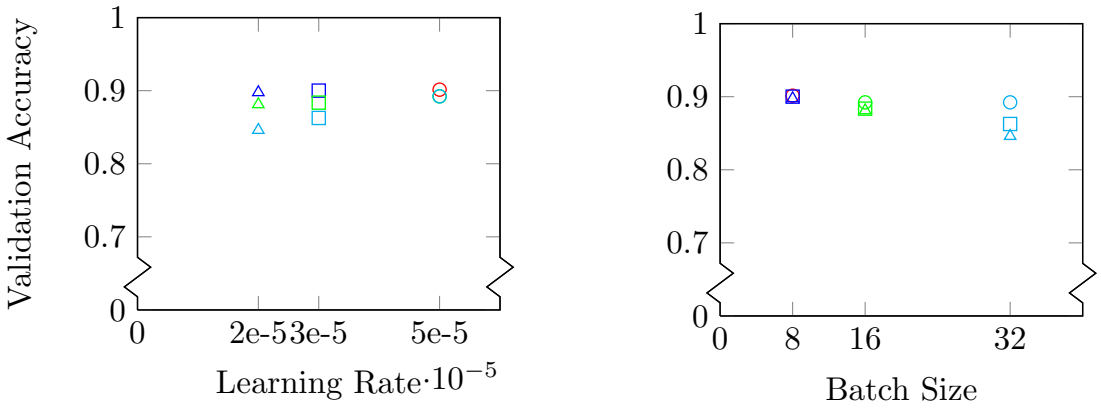

Figure 4.5: LEGAL-BERT hyperparameter optimisation results

seems to increase as the learning rate increases and the batch size decreases. Most models performed best with a learning rate of 5e-5 and a batch size of 8, and for all models, the validation accuracy was the lowest with a learning rate of 2e-5 and a batch size of 32. The higher validation accuracies of the learning rates on the higher end of our range suggest that the learning rate is not yet too large such that it converges too quickly to a suboptimal solution. Moreover, the larger learning rate helps to reduce training time. On the other hand, the increase in validation accuracy as the batch size decreases can be attributed to the fact that the models are able to update the model weights more often with smaller batch sizes. These frequent updates based on small subsets of the data introduce some noise into the system as the smaller batches result in more divergent estimates of the gradient. This noise allows the model to explore more directions in the solution space, making it less likely to end up in a local optimum and more likely to converge to the global optimum.

It is interesting to note that the pattern we identified for the hyperparameter settings for the models fine-tuned on the English dataset does not persist in the results for the grid search for Multilingual BERT, which was fine-tuned on the larger dataset with Dutch sentences. The results for this model are shown in figure 4.6. The optimal combination is a learning rate of 2e-5 and a batch size of 16. Whereas we cannot be certain as to why Multilingual BERT breaks this pattern that was observed for the other models, it could be attributed to the differences in the datasets the models were fine-tuned on as they contain entirely different languages with their own characteristics. For each model, we selected the best combination based on the results from the sweeps. The final fine-tuning settings that we used to train our models are provided in figure 4.3. We fine-tuned each model for four epochs, and we used a weight decay of 0.01 to prevent overfitting.

4.2.2 Fine-tuning losses

To monitor the performance of the models during fine-tuning, we kept track of the cross-entropy loss, which quantifies the difference between the true and the predicted probability distributions across the classes.

For the models that we fine-tuned on the English dataset, we logged the training and validation loss every 10 training steps. We logged these losses of the M-BERT

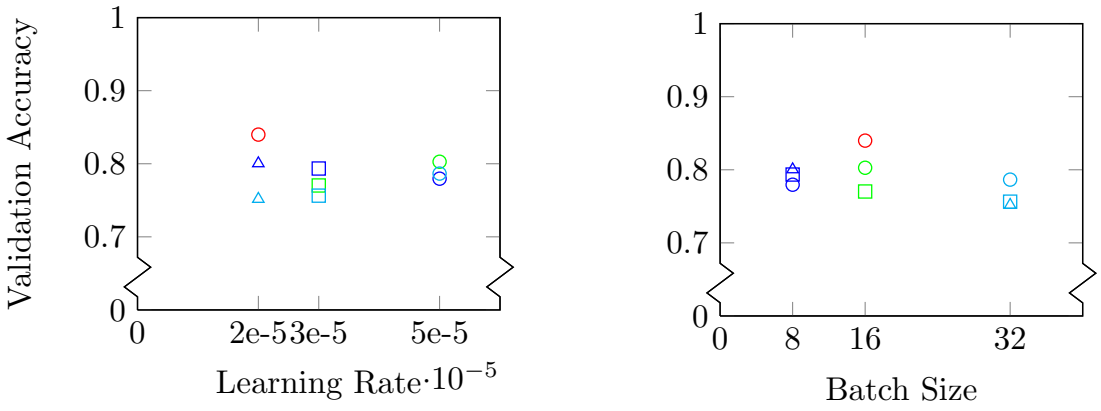

Figure 4.6: M-BERT hyperparameter optimisation results

Table 4.3: The hyperparameter settings for the fine-tuning of the final models, based on the results from the hyperparameter optimisation.

Model	Learning rate	Batch size
BERT	3e-5	8
Multilingual BERT	2e-5	16
SpanBERT	5e-5	8
LEGAL-BERT	5e-5	8
EURLEX-LEGAL-BERT	5e-5	8

model, fine-tuned on the Dutch dataset, every 50 steps.

Figure 4.7 shows the fine-tuning losses for the models fine-tuned on the English dataset. For each model, we plot the training loss and the validation loss in the same graph over all 616 training steps. We see that for all models, the training loss decreases the most in the first 100 steps after which the curves start to flatten, showing that the models learn the most during these first 100 steps. We also observe that for all models, after about 300 training steps the validation loss remains the same or starts to slightly increase, whereas the training loss continues to decrease. This tells us that after this point, the models are overfitting.

Figure 4.7 does not allow us to effectively compare the losses of the different models. Therefore, we combined the training losses and validation losses of the models in figure 4.8. The training losses are shown in the left plot and the validation losses are displayed in the plot on the right. We see that all the models follow the same pattern for both the training and validation loss. The losses for BERT, LEGAL-BERT, and EURLEX-LEGAL-BERT are very close together, but the training loss of SpanBERT is slightly higher than the other models. However, SpanBERT's validation loss remains flat after 300 training steps, whereas the validation losses of the other models slowly increase. This may indicate that SpanBERT is generalising better on the fine-tuning data. In section 4.3, we compare the predictions of the models to find out if that is correct.

Whereas we also fine-tuned M-BERT for four epochs, the model had more steps because of the larger size of the dataset. Another difference is that we used a batch size of 16 for M-BERT. As a result, the model was fine-tuned for 864 steps. The training and validation loss for M-BERT is shown in figure 4.9. M-BERT's

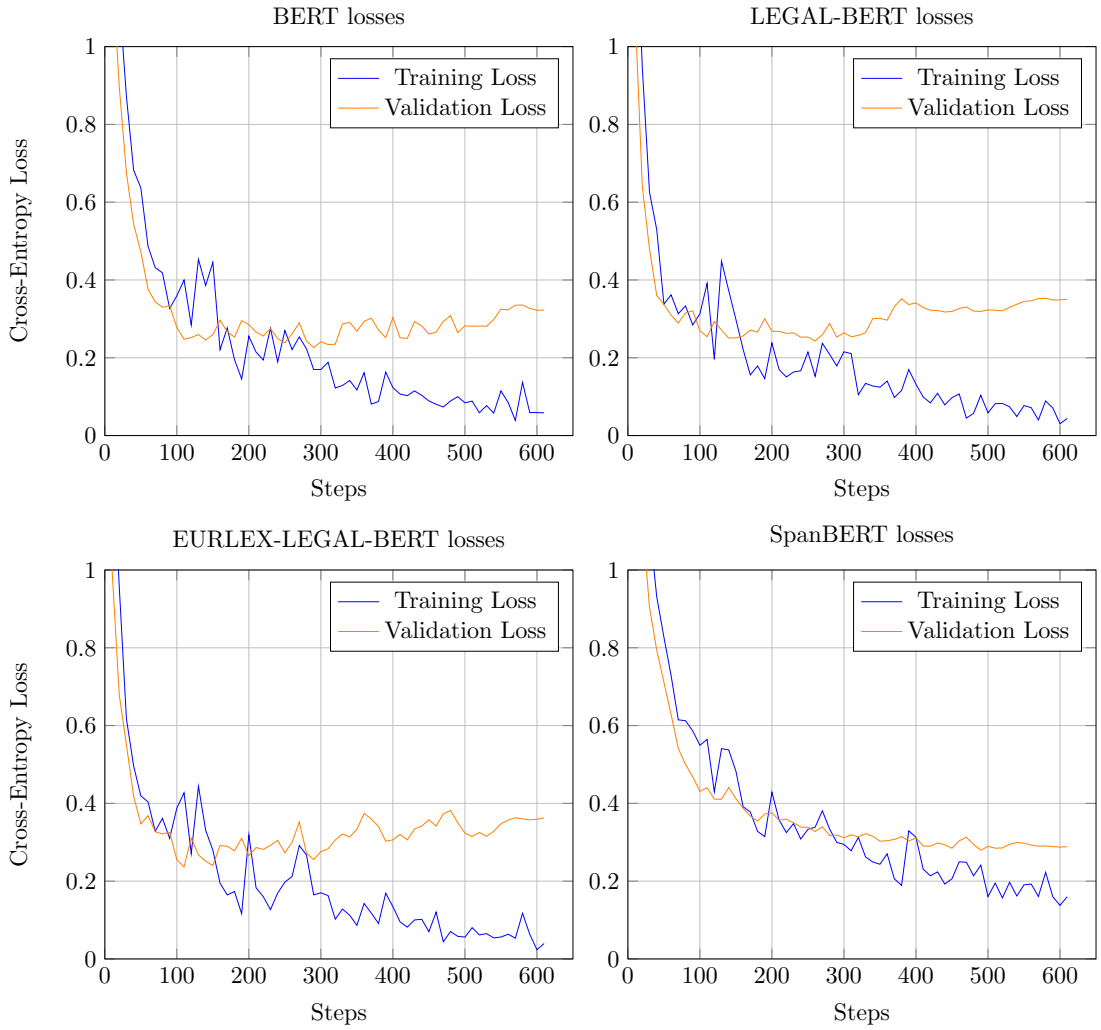

Figure 4.7: The training and validation losses of the models that were fine-tuned on the English dataset.

training process exhibits a similar pattern to that of the other models, learning the most in the first 100 training steps and starting to overfit after about 300 steps. However, where the validation losses of the other models plateaued around 0.3, M-BERT's validation loss levels out at around 0.5. This indicates that M-BERT is not performing as well as the other models. We must note, however, that the models are not validated on the same sets of data (M-BERT's validation set also consists of Dutch sentences). As such, we cannot draw any conclusions from these results and we compare the predictions of the models in the next section.

4.3 Evaluation of all model predictions

In this section, we present and evaluate the predictions made by our models on the test set. In section 4.3.1, we describe our evaluation of the predictions of the model for the individual tokens in our dataset. We also consider how well the models manage to identify spans of tokens corresponding to the Flint roles, for which we report the results in section 4.3.2.

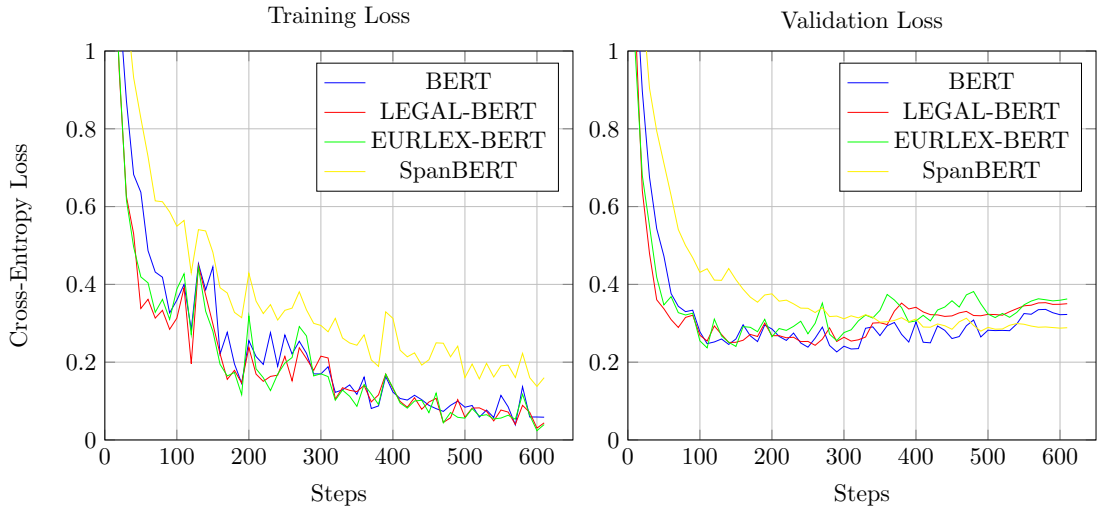

Figure 4.8: The losses of all of the models that were fine-tuned on the English dataset. The training and validation losses are plotted in the left and right figures, respectively.

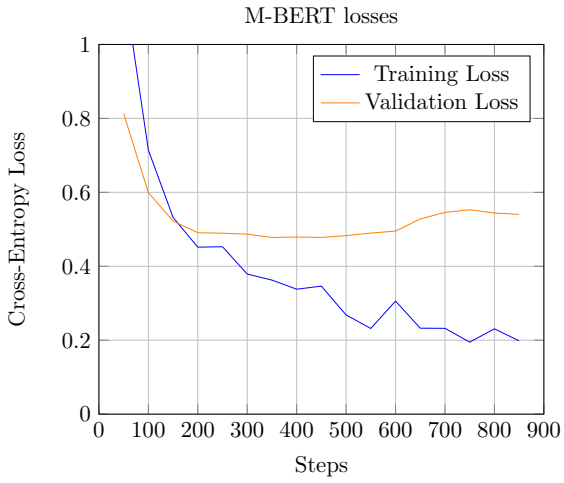

Figure 4.9: The training and validation losses of Multilingual BERT, which was fine-tuned on the Dutch dataset from Van Drie et al. (2023).

4.3.1 Token classification

As was discussed in section 3.3, we computed the accuracy, balanced accuracy, precision, recall, the macro, and weighted F1-score, as well as the F1-score per Flint role to analyse and compare how well the models performed at labelling individual words in the action sentences with the Flint act labels.

The scores for the first six metrics are presented in table 4.4. When comparing the scores for all of the models, we notice that the rule-based baseline model yields the worst scores on all six of the reported metrics. Whereas the mapping model performs significantly better than the rule-based baseline, it does not outperform any of the BERT-based models on any of the metrics. The results for the BERT models fine-tuned on the English dataset are all very close together. The Multilingual BERT model that was fine-tuned on the Dutch dataset has a lower performance than the other BERT model, but the results are still well above the rule-based and

mapping approaches. SpanBERT's scores are very close to BERT's results, but it performs slightly worse on all metrics. Moreover, we see that for the fine-tuned models the precision and recall are relatively close, but for the rule-based baseline and the mapping model the recall is much lower than the precision. This indicates these models yielded a lot of false negatives.

Overall, we observe that LEGAL-BERT and EURLEX-LEGAL-BERT, the models pretrained on data from the legal domain obtain the highest scores. LEGAL-BERT reports the highest balanced accuracy and F1-score of 89.2% and 90.1%, respectively. EURLEX-LEGAL-BERT yields the highest values on accuracy and weighted F1, with 89.6% on both metrics.

Table 4.4: Results of the evaluation on the token-level

	Rule-based	**Mapping**	**BERT**	**M-BERT**	**SpanBERT**	**LEGAL-BERT**	**EURLEX-BERT**
Accuracy	0.528	0.737	0.879	0.829	0.880	0.881	**0.896**
Balanced accuracy	0.464	0.611	0.885	0.890	0.890	**0.892**	0.885
Precision	0.551	0.795	0.888	0.830	0.880	0.911	**0.917**
Recall	0.464	0.611	0.885	0.832	0.890	0.892	**0.896**
Macro F1	0.433	0.666	0.886	0.829	0.884	**0.901**	0.900
Weighted F1	0.473	0.729	0.881	0.830	0.880	0.881	**0.896**

Figure 4.5 shows the F1-scores per Flint role for all of the models. This overview provides better insight into how well the models are able to classify the individual roles in the role labelling task. The rule-based baseline generally has the worst performance for all roles, with the exception of the *action* role, on which the mapping model scores lower. In section 4.3.2 we provide some examples of labelled sentences that can shed light on this observation.

Once again, the mapping model generally improves upon the rule-based model but does not come close to the results of the models fine-tuned on Flint roles specifically. Additionally, the LEGAL-BERT and EURLEX-LEGAL-BERT yield the highest scores in all categories, supporting our hypothesis that domain-specific models can increase performance on this task.

For all models, we notice that they obtain the highest scores on the *action* and *actor* roles — apart from the mapping model which scored low for the *action* role — and that they all score lower for the *object* and *recipient* roles. This is a result that we expected given the lower inter-annotator agreement for the *object* and *recipient* as discussed in section 4.1.1, and the class imbalances discussed in section 4.1.2.

Table 4.5: The F1 scores per role label for all models

	Rule-based	**Mapping**	**BERT**	**M-BERT**	**SpanBERT**	**LEGAL-BERT**	**EURLEX-BERT**
Action	0.678	0.536	0.972	0.915	0.962	0.978	**0.983**
Actor	0.567	0.847	0.955	0.927	0.956	**0.957**	0.938
O	0.623	0.770	0.898	0.849	0.899	0.893	**0.914**
Object	0.282	0.725	0.829	0.779	0.827	0.821	**0.850**
Recipient	0.016	0.454	0.777	0.678	0.776	**0.854**	0.813

In figure 4.10 the normalised confusion matrices for the rule-based, mapping, BERT, and M-BERT models are shown, which contain the accuracies per class.

The figure provides a more intuitive visualisation of how the models performed. We clearly see that the rule-based model is relatively successful at recognising the *action* in a sentence, but performs poorly for the *actor* and *recipient*. The mapping model does well on classifying the *actor* tokens but has more problems with the *action* and *recipient*. In the matrices for these two models, it seems that they often default to classifying tokens as *other*. As a result, these models also report a high accuracy for the *other* class, and considering the over-representation of *other* tokens in the annotated dataset, we can see how this may have caused a bias in the reported accuracies in table 4.4 and why we should pay more attention to the balanced accuracy.

The confusion matrices for BERT and M-BERT indicate that most roles are classified correctly most of the time. The most common mistake made by the systems is that they classify words whose true label is *recipient* as *object*. The confusion matrices for LEGAL-BERT, EURLEX-LEGAL-BERT, and SpanBERT are very similar to that of BERT and are included in appendix A.4. confusion matrices

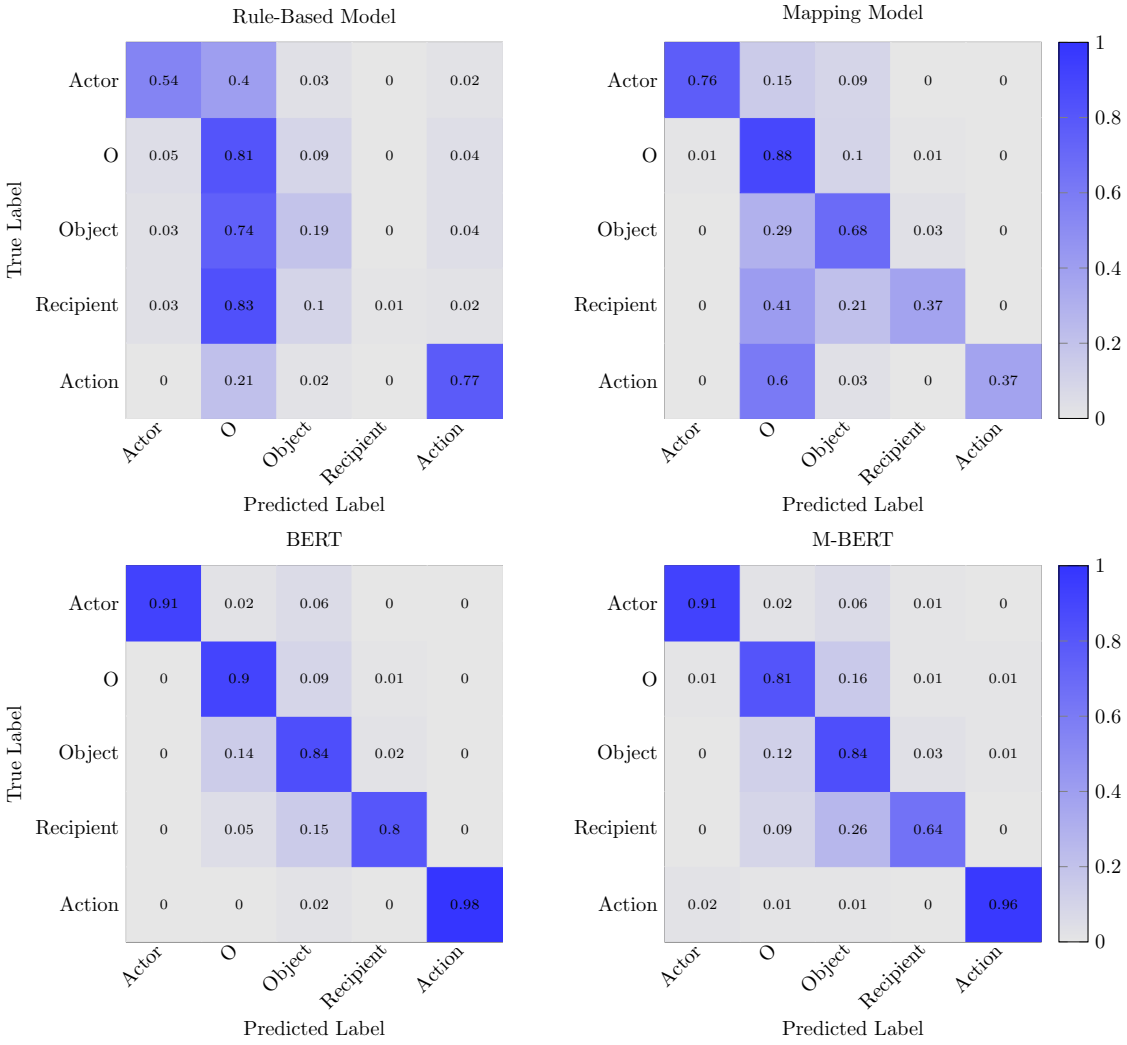

Figure 4.10: The mean confusion matrices for the rule-based model, the mapping model, BERT, and M-BERT.

4.3.2 Role classification

Given that we are working towards a system that can automatically extract entire roles from legal text that can be used to fill Flint act frames, we used the MUC evaluation approach used for the SemEval-2013 task (Segura-Bedmar et al., 2013) described in section 3.3 to determine whether our systems identified a complete or partial role and whether they assigned the correct role type. This section presents the results of that evaluation.

In table 4.6 the overall F1 scores for the MUC evaluation, which do not differentiate between the different Flint roles, are reported for all the models on all four of the different evaluation schemas. The numbers based on which these F1 scores have been computed are shown in appendix A.5.1, which shows the counts of correct, incorrect, partial, missed, and spurious roles.

These results exhibit nearly the same pattern as the results for the evaluation on the token level: the rule-based model consistently yields the lowest scores in all categories and the mapping model improves upon the rule-based baseline but does not come close to the results of the models that were fine-tuned on Flint roles and once more, the M-BERT model scores lower on all categories than the models fine-tuned on the English dataset and the domain-specific language models have the best performance, with LEGAL-BERT specifically scoring the highest on all schemas. We note that the scores for BERT and SpanBERT are again very close, with SpanBERT obtaining slightly better results on all schemas.

All models obtain their highest F1-score for the *type* schema, which only considers whether a model assigned the correct role type to an identified role, regardless of how it has identified the string boundaries of that role. The lowest F1-scores are consistently observed for the *strict* evaluation schema, which requires a model to have an exact match on the boundaries of the identified role as well as the Flint role type assigned to that role. On this schema, LEGAL-BERT denotes the highest F1-score at 82.2%.

Table 4.6: Overall F1 scores on the MUC evaluation for all the models

	Rule-based	**Mapping**	**BERT**	**M-BERT**	**SpanBERT**	**LEGAL-BERT**	**EURLEX-BERT**
Type	0.601	0.788	0.906	0.798	0.910	**0.924**	0.917
Partial	0.527	0.671	0.866	0.732	0.873	**0.888**	0.855
Exact	0.389	0.499	0.802	0.629	0.812	**0.833**	0.830
Strict	0.375	0.475	0.790	0.614	0.798	**0.822**	0.815

Table 4.7 provides a closer look into the F1 scores on the different evaluation schemas per role. Once again the correct, incorrect, partial, missed, and spurious role counts to compute these values have been reported in the tables in appendix A.5.2. When we compare the models, we see that LEGAL-BERT generally has the best overall performance, followed closely by EURLEX-LEGAL-BERT and SpanBERT, with EURLEX-LEGAL-BERT obtaining the best results for the *action* role and SpanBERT matching LEGAL-BERT for the *object* and *recipient* roles. Similarly to what we saw in the evaluation of the models on the token level in section 4.3.1, the models generally obtain the highest scores for the *actor* and *action*. Furthermore, we notice that for all models the difference in scores for the *partial* schema and the

exact schema, which requires an exact boundary match on the role, is lowest for the *actor* and *action* indicating that the models were more frequently successful in identifying these roles in their entirety. Looking at the *exact* scores for the *object* and *recipient*, it seems that getting the correct boundaries for these roles was a bigger challenge for the models.

In the sections below, we will go into some specific observations for the individual models and provide examples of sentences that were labelled by the models.

Table 4.7: The F1 scores per role per evaluation schema for all the models. In the schema column, the T, P, E, and S represent the *type*, *partial*, *exact* and *strict* evaluation schemas, respectively.

Role	Schema	Rule-based	Mapping	BERT	M-BERT	SpanBERT	LEGAL-BERT	EURLEX-BERT
Actor	T	0.72	0.89	**0.97**	0.90	0.96	0.96	0.95
	P	0.65	0.90	0.97	0.88	0.97	**0.98**	0.96
	E	0.56	0.86	0.94	0.83	0.95	**0.96**	0.94
	S	0.54	0.83	0.92	0.80	0.93	**0.94**	0.91
Action	T	0.67	0.78	0.95	0.91	0.94	**0.97**	**0.97**
	P	0.61	0.45	0.93	0.84	0.89	**0.94**	**0.94**
	E	0.52	0.09	0.89	0.76	0.84	0.90	**0.92**
	S	0.52	0.09	0.89	0.76	0.84	0.90	**0.92**
Object	T	0.51	0.76	0.84	0.62	**0.85**	**0.85**	**0.85**
	P	0.37	0.72	0.75	0.53	**0.78**	**0.78**	0.77
	E	0.15	0.63	0.63	0.38	**0.68**	0.67	0.66
	S	0.15	0.60	0.62	0.37	**0.66**	**0.66**	0.65
Recipient	T	0.03	0.59	0.75	0.69	**0.85**	**0.85**	0.82
	P	0.33	0.68	0.71	0.66	**0.80**	0.78	0.78
	E	0.22	0.59	0.60	0.53	**0.71**	**0.71**	0.69
	S	0.03	0.49	0.58	0.49	0.67	**0.71**	0.65

4.3.2.1 Observations for the rule-based model

In table 4.7 it is shown that the rule-based model obtains very low scores for the *object* and *recipient* roles. We see that the model yields an F1 of 37% in the *partial* schema of the *object* role, indicating that it does not do well at grasping the boundaries of *objects*. Moreover, we note that the model has an F1 of 3% for the *type* schema of the *recipient*, meaning that the model almost never assigns the correct type to roles that have the *recipient* label in the true annotation. Examples of these observations are provided in figure 4.11. In the example, we see that the rule-based model does recognise part of the object, but does not get the correct boundaries. Also, we see that the model has included the word "them" in the action, whereas it is annotated as the *recipient* in the true annotation.

4.3.2.2 Observations for the mapping model

The F1 scores for the mapping model indicate that the mapping model performs worse at identifying the *action* in sentences than the rule-based model. This can be explained by the fact that the mapping, as provided in section 3.2.2, only maps the root of the sentence to the action, which is always just one word. This means it does not map any auxiliary verbs such as *shall* or *may* to the action. As a result,

Actor - Action - Object - Recipient

True annotation:
The Commission may order them to provide access to, and explanations relating to, its databases and algorithms.

Rule-based prediction:
The Commission may order them to provide access to, and explanations relating to, its databases and algorithms.

Mapping model prediction:
The Commission may order them to provide access to, and explanations relating to, its databases and algorithms.

Figure 4.11: An example sentence from the Digital Service Act, taken from the test set, with the true annotations and the predictions by the rule-based baseline and the mapping model.

we see that the mapping model has a decent F1 score of 78% for the *type* schema, which only cares about assigning the correct type to a role and does not consider the boundaries. However, for the *exact* and *strict* schemas, which do require an exact boundary match, the F1 is only 9%. In figure 4.11, we can see how the model only labels the root of the sentence as the action and fails to include the word *may*.

4.3.2.3 Observations for the fine-tuned models

Similar to the rule-based baseline, the fine-tuned models show a significant decrease in the scores for the *object* and *recipient* roles. On the *strict* evaluation schema, the models that were fine-tuned on the English set yield an F1 of at most 66% for the *object* and 71% for the *recipient*. M-BERT scores even lower, 37% and 49% for the *object* and *recipient*, respectively. However, the scores for these models on the *type* and *partial* evaluation schemas are much higher. This suggests that the models are doing relatively well at identifying a part of the tokens that are assigned as these roles in the true annotation and assigning the correct role type to these tokens, but that they struggle to get the boundaries right. Out of all the fine-tuned models, SpanBERT does the best at identifying roles in their entirety, with F1 scores of 68% and 71% for the *object* and *recipient* on the *exact* evaluation schema, which considers whether the models achieved an exact boundary match. M-BERT's drop in performance for these roles suggests that as the roles become longer, and the language within them more complex, it becomes harder to transfer the knowledge from Dutch to English. In figure 4.12 we can see an example of a sentence from the test set where almost all models failed to get the boundaries of the *object* right. We do notice, however, that all models managed to identify a considerate portion of the true *object*. In this example, only SpanBERT assigned the correct boundaries and type to the role.

Another observation is the improvement that LEGAL-BERT, EURLEX-LEGAL-BERT, and SpanBERT make on the *recipient* role in comparison to BERT. For all other roles, the performance of BERT is very close to the other models fine-tuned on the English set, but for the *recipient* the different is much larger. Figure 4.13 provides an example of this observation, where LEGAL-BERT and EURLEX-BERT match the true annotation, but BERT confuses the *object* and *recipient*.

Actor - Action - Object - Recipient

True annotation:
Providers of very large online platforms and of very large online search engines shall put in place reasonable, proportionate and effective mitigation measures, tailored to the specific systemic risks identified pursuant to Article 34, with particular consideration to the impact of such measures on fundamental rights.

BERT prediction:
Providers of very large online platforms and of very large online search engines shall put in place reasonable, proportionate and effective mitigation measures, tailored to the specific systemic risks identified pursuant to Article 34, with particular consideration to the impact of such measures on fundamental rights.

M-BERT prediction:
Providers of very large online platforms and of very large online search engines shall put in place reasonable, proportionate and effective mitigation measures, tailored to the specific systemic risks identified pursuant to Article 34, with particular consideration to the impact of such measures on fundamental rights.

SpanBERT prediction:
Providers of very large online platforms and of very large online search engines shall put in place reasonable, proportionate and effective mitigation measures, tailored to the specific systemic risks identified pursuant to Article 34, with particular consideration to the impact of such measures on fundamental rights.

LEGAL-BERT prediction:
Providers of very large online platforms and of very large online search engines shall put in place reasonable, proportionate and effective mitigation measures, tailored to the specific systemic risks identified pursuant to Article 34, with particular consideration to the impact of such measures on fundamental rights.

EURLEX-LEGAL-BERT prediction:
Providers of very large online platforms and of very large online search engines shall put in place reasonable, proportionate and effective mitigation measures, tailored to the specific systemic risks identified pursuant to Article 34, with particular consideration to the impact of such measures on fundamental rights.

Figure 4.12: An example sentence from the Digital Service Act, taken from the test set, with the true annotations and the predictions by the fine-tuned BERT models.

Actor - Action - Object - Recipient

True annotation:
If that authority so requests, its officials may assist the officials and other accompanying persons authorised by the Commission to conduct the interview.

BERT prediction:
If that authority so requests, its officials may assist the officials and other accompanying persons authorised by the Commission to conduct the interview.

M-BERT prediction:
If that authority so requests, its officials may assist the officials and other accompanying persons authorised by the Commission to conduct the interview.

LEGAL-BERT prediction:
If that authority so requests, its officials may assist the officials and other accompanying persons authorised by the Commission to conduct the interview.

EURLEX-LEGAL-BERT prediction:
If that authority so requests, its officials may assist the officials and other accompanying persons authorised by the Commission to conduct the interview.

Figure 4.13: An example sentence from the Digital Markets Act, taken from the test set, with the true annotations and the predictions by BERT, LEGAL-BERT, and EURLEX-LEGAL-BERT

5 Discussion

In this thesis, we set out to explore a system that can automatically label English legal text with the four semantic roles associated with the Flint act frame. In this chapter, we conclude our work and discuss our findings. We start in section 5.1 by answering the research questions that we posed at the beginning of this study. Next, we delve into the limitations of our method in section 5.2. We discuss some of our interpretations of the results achieved by the models in section 5.3. In section 5.4, we suggest several directions for future work and we finish in section 5.5 by discussing the relevance of this work.

5.1 Conclusion

The main question raised in this thesis was: *How can we automatically label the roles of Flint act frames in sentences from English legal text?*

We specified several sub-questions to guide us towards an answer to our main research question. We start off this section by answering these sub-questions, after which we will formulate our answer to the main research question.

1. *Which methods already exist for semantic role labelling?*

After conducting an extensive review of the relevant literature, we identified numerous methods that have been widely employed to tackle the task of semantic role labelling. Some of the earliest SRL systems consisted of rule-based approaches, using syntactic parsing, often in combination with dependency parsing, and applying sets of (manually crafted) rules to identify and label the arguments in sentences. However, a prevalent limitation for these kinds of systems is that they heavily depend on the quality of their rules, making them less reliable due to the complexity of natural language and less generalisable as rules may need to be updated for different domains. To avoid this limitation, later systems used supervised machine learning approaches such as SVMs and, even more recently, neural approaches such as BiLSTMs to improve upon the performance of rule-based systems. The most recent successes in semantic role labelling can be attributed to systems that utilise pretrained transformer-based language models and fine-tune them on smaller sets of data that have been annotated specifically for the intended SRL task, as they better capture the contextual dependencies in natural language. Most notably, BERT has achieved state-of-the-art results on SRL benchmarks.

2. *How do existing semantic role labelling roles relate to Flint act frame roles?*

To determine the relation between Flint act frame roles and existing semantic roles, we first studied the literature on the different types of semantic roles. Numerous forms of semantic roles can be distinguished, such as the roles associated with FrameNet, which are specific to the slots of unique semantic frames, and the roles from the Proposition Bank, which are numbered from one to six but have a specific function for individual verb senses. Some of the most commonly known semantic roles are thematic roles, which are general roles that provide a human-readable description of how arguments and predicate relate. Work by Van Drie et al. (2023), pointed out the similarity between Flint roles and thematic roles, as both are human-readable and convey the abstract relationship of the argument to the predicate without going into the specific semantic interpretation of the argument. They provided an initial mapping of several thematic roles to Flint's act frame roles. In this work, we extended this mapping by comparing VerbAtlas' definitions of thematic roles to the function of Flint act frame roles.

To test how well a pretrained SRL model could label sentences with Flint act frame roles, we took an existing pretrained BERT model that labelled sentences with PropBank roles and mapped these roles to thematic roles and subsequently to Flint roles. This model achieved a significant improvement over the rule-based baseline, with a weighted F1 score of 72.9% when evaluated on the token level, and scores of 67.1% and 47.5% on the MUC *partial* and *strict* evaluation schemas, respectively.

3. *How can existing language models and techniques such as fine-tuning contribute to labelling Flint roles?*

In the exploration to answer subquestion 1, fine-tuning BERT was identified as one of the most effective methodologies for semantic role labelling tasks in the current academic field. Based on this result, we opted to explore the potential of fine-tuning BERT to label text with Flint roles in this thesis and hypothesised that it would lead to improved performance compared to the rule-based baseline and the mapping model of the previous subquestion. The rule-based model achieved a weighted F1 of 47.3% on the token-level evaluation and F1 scores of 52.7% and 37.5% on the MUC's *partial* and *strict* evaluation schemas, respectively.

In order to fine-tune our models, we conducted a thorough annotation process on action sentences from several EU regulations. This resulted in a dataset of 1573 annotated sentences, which is a significant contribution of this thesis.

After fine-tuning BERT on this dataset, the evaluation of the model showed a substantial improvement over both the rule-based baseline and the mapping model of the previous subquestion, with a weighted F1 score of 88.1% on the token-level evaluation and an F1 of 79% on MUC's *strict* schema.

4. *What effect does using a domain-specific or task-specific language model have?*

We hypothesised that fine-tuning BERT models that have been pretrained on text from the legal domain or that were pretrained to model spans of text would improve upon BERT's results on the defined task of labelling legal text with semantic roles. To test this hypothesis, we fine-tuned LEGAL-BERT, EURLEX-LEGAL-BERT, and SpanBERT on our annotated dataset.

After the evaluation of the models, we found that all of these models achieved improved performance on the task in comparison to BERT. On the token-level evaluation, EURLEX-LEGAL-BERT yielded the highest scores. On the MUC evaluation, all models consistently outperformed BERT on all schemas, with LEGAL-BERT yielding the overall highest scores on all schemas. On the *strict* MUC evaluation schema, LEGAL-BERT showed an overall improvement over BERT over 4%. These results support our hypothesis and, hence, we conclude that there is an added value to using these domain-specific and task-specific language models on this particular task, with especially the domain-specific language models showing promising results.

5. *Can we generalise a language model trained to label Dutch legal text to label English legal text?*

Given the availability of the larger dataset with sentences from Dutch laws, we experimented with how we could leverage this dataset for labelling English sentences. Based on the results from the reviewed literature, which showed that M-BERT is successful at generalising across languages, we hypothesised that we would see results similar to or even better than BERT given the larger size of the Dutch dataset. To test this hypothesis, we fine-tuned M-BERT on the dataset by Van Drie et al. (2023), which provides annotations of Flint roles on 4463 sentences from Dutch laws. After fine-tuning on the Dutch dataset and evaluating on the English test set, we found that whereas M-BERT outperforms the rule-based baseline and the mapping model, it does not manage to reach the level of BERT on our task. It shows that M-BERT does obtain F1 scores of over 90% for the *actor* and the *action*, but struggles with the *object* and *recipient* roles, from which we conclude that more language-specific fine-tuning is required to learn these longer and more complex roles in legal text.

Finally, we return to the main question posed in this thesis: *how can we automatically label the roles of the Flint act frames in sentences from English legal text?*

To answer this research question, we implemented seven different models: a rule-based model, a model that maps existing semantic roles to Flint roles, four variations of the BERT model fine-tuned on a dataset with Flint role annotations on English legal text, and a Multilingual BERT model fine-tuned on a dataset with annotations on Dutch legal text. These models were evaluated and compared on their ability to correctly assign words in legal sentences to Flint act roles, as well as their ability to correctly identify (parts of) roles. Furthermore, we contributed a dataset of 1573 sentences from EU regulations with Flint act role annotations. After considering all of our results and the answers to the subquestions defined above, we conclude that fine-tuning a domain-specific language model is the most successful approach for the task specified for this thesis.

5.2 Limitations

In this section, we reflect on our methodology and results while discussing their limitations. Hereby, we aim to uncover and address any constraints or biases that may have impacted the results of our work for this thesis, and the implications they may have.

5.2.1 Annotation process

The annotation process plays a crucial role in the way we can interpret the results that we obtained in this thesis. The quality of the annotations directly impacts how well the fine-tuned models can learn and how suitable the test set is for the evaluation of all models.

During the annotation process, a subset of 200 sentences was selected for inter-annotator agreement evaluation. In the current setup, all annotators were presented with these sentences in a predetermined order, before they were shown their additional unique set of sentences to be annotated. Upon further consideration, it would have been preferred to randomly distribute these sentences for annotator-agreement evaluation over the total batch of sentences to be annotated, to avoid a pattern of learning in the annotations that could result in systematic biases.

In the evaluation of the inter-annotator agreement per role in section 4.1.1, it was shown that among annotators there was a lower — yet sufficient — agreement for the *object* and *recipient* roles. For all models, this same pattern can also be recognised in the results of their predictions on the test set. These observations suggest that the final dataset had a less consistent and objective reference for the correct *object* and *recipient* (at least not on the same level as the *actor* and *action*), which may have impacted the results in two ways: the fine-tuned models may have had more difficulty to learn these roles, and there may have been a lack of a consistent standard in the test set. To mitigate this limitation, we would need to reconsider our instructions in the annotation protocol for these roles as our definition for these roles may have been too narrow or too wide. Moreover, we could consider extending the number of examples and practice sentences. Lastly, we note that despite improvements in the instructions, these roles may simply be more challenging for annotators to recognise.

Whereas the results of our evaluation of inter-annotator agreement show that annotators have more trouble with the *object* and *recipient* roles, the metrics that we chose for evaluation only analyse the agreement and the token level and do not provide a clear insight into exactly *what* the annotators are disagreeing on. It would be useful to incorporate a method into our evaluation that provides insight into whether annotators disagreed on the role type, the boundaries of the role, or the role in its entirety, as it would allow us to understand in what way we need to update the annotation protocol. For this reason, we believe that implementing an additional metric for inter-annotator agreement that measures agreement over spans of text could improve our current method. However, at the moment we were not able to find a universal metric to judge the agreement on spans of text.

We also note that because of the way the annotation tool is designed, we are not able to distinguish between different occurrences of the same role within sentences. In view of the ultimate goal of filling Flint frames, this poses a significant limitation. Sentences may contain multiple *actors*, *actions*, *objects* and *recipients*, and the current setup of the tool does not allow us to distinguish between roles or link certain roles to the same action. When filling Flint frames, this information is crucial. As such, future work could benefit from an update to the annotation tool that allows for the explicit linking of roles to certain actions.

As a last remark about the dataset in this thesis, we recognise that the performance of the models might improve if there is more data to fine-tune on. As such, an easy direction for future work is to annotate additional sentences, potentially from other sources of laws to make the models more generalisable.

5.2.2 Setup of rule-based baseline

The results of the rule-based model showed the lowest performance of all models. Additionally, the results indicated that the model only assigned the *recipient* role label to a limited number of roles. Despite efforts, we were not able to discover any specific flaw in our implementation that could account for this observation. However, it might be possible that the POS tags or dependency tags that are associated with the *recipient* role are rarely assigned in our set of sentences. Due to time constraints, we were not able to carry out this analysis in this work. Future work could look more closely into the underlying reasons for the model's behaviour for this role and update its rule accordingly.

This specific observation for the *recipient* role highlights a larger characteristic of the rule-based model, which is that its effectiveness is directly tied to the quality and completeness of the POS-tagger, the dependency parser, and the rules that it relies upon. As we saw in table 4.4 in section 4.3.1, the recall of the rule-based baseline was very low, which is indicative of a lot of false negatives. Therefore, we believe that updating and further specifying the rules of this model could lead to significant jumps in performance since new rules might be able to capture the roles that the current model missed. However, it is important to keep in mind that as the rules become too detailed, the model might lose its ability to generalise across the legal domain, beyond just EU regulations. As such, in order to create a system that can function in multiple areas of the legal domain, for future work it would be interesting to focus on creating rules that strike the balance between being detailed and generalising well across different legal texts.

5.2.3 Setup of the mapping model

In the results of the mapping model, we observed that its performance lacks the most with the *action* role. As was briefly touched upon in section 4.3.2.2, this can be explained by the fact that the mapping model only maps the word that is labelled as the root of each action sentence to the Flint *action* label, disregarding auxiliary verbs such as *shall* or *may*. This is reflected in the results, as the model often gets a partial match on the role, but hardly ever an exact match. To mitigate this limitation, the mapping for the *action* would need to be updated.

From this, it follows that similar to how the results of the rule-based model depend on the quality of its rules, the results of the mapping model depend on the quality of the different components on which it is based. Our model distinguished three components: the pretrained SRL model that labelled text with PropBank roles, the mapping from PropBank to thematic roles, and the mapping from thematic roles to Flint roles.

First, it should be noted that the mapping model is inherently limited to the ability of the SRL model that initially labels the sentences. The `transformer-srl` model yielded an F1 of 86% on the CoNLL 2012 set, so we could not expect our final model to achieve much higher results, especially since the SRL model was trained on out-of-domain texts.

Apart from the underlying SRL model, the model depends on the mappings for the different forms of semantic roles. Whereas the VerbAtlas resource for mapping PropBank roles to thematic roles is quite complete with a mapping for over 5000 verbs, the mapping from thematic roles to Flint roles is less robust. This limitation

mainly stems from the fact that the definitions for both thematic roles and Flint roles are not universally defined. As a result, this more unstable mapping can have adverse effects on the predictions made by the model and its ability to extract full roles. Additionally, the fact that the model relies on multiple mappings makes it more unstable. Each separate component makes the model more complex and makes it more difficult to identify the origin of mistakes in the predictions. For this reason, a simpler model with fewer components would be preferable.

An interesting alternative to our current mapping model could be a model that is directly fine-tuned to label (legal) text with thematic roles and map its resulting roles to Flint. Unfortunately, at the time of this research, we are not aware of the existence of any pretrained SRL models that use thematic labels in the legal domain. Given that one of the main advantages of the mapping model is that it does not require annotated data, we see no added value in creating such a model for the specific purpose of subsequently labelling Flint roles.

5.2.4 Setup of the fine-tuned models

The results of the hyperparameter optimisation for the fine-tuned models revealed that, in general, the performance of our models increased with smaller batch sizes. The smallest value we experimented with, which often achieved the highest validation accuracy, was a batch size of 8. Moreover, the results showed that most models yielded the best validation accuracy with the largest learning rate out of our selected range (5e-5). Considering that with our current ranges the performance is still increasing with smaller batch sizes and larger learning rates, we believe that it might be valuable to experiment with larger ranges for these hyperparameters in future work, where we would consider implementing smaller batch sizes and larger learning rates. However, adding more values to the grid search would require a lot of extra training time. To find the optimal values for the hyperparameters but keep fine-tuning time to a minimum, we could consider implementing a more efficient search method such as a Bayesian search, which learns from previous iterations and, consequently, does not require us to test the entire sample space.

Apart from experimenting with more values for the batch size and learning rate, there are some other model settings that could be explored further in future work. As we saw in figure 4.7 in section 4.2.2, most of the models started to overfit after approximately 300 training steps. To avoid the occurrence of overfitting in the models, we could consider implementing several techniques such as regularisation, early stopping, or cross-validation. Moreover, we believe future work on these models would benefit from annotating additional sentences to allow the models to train on more data, which could also help mitigate overfitting.

Another benefit of adding more data to the dataset is that it may help the fine-tuned model to become more generalisable to other legal texts. Showing the models more data from different laws will help it to understand more types of *actors*, *actions*, *objects*, and *recipients*, as each law has its own set of acts that it describes. Moreover, adding annotated sentences from legal sources other than European Union regulations would result in even more general systems. In this case, we would expect that the LEGAL-BERT model would be the most effective approach. EURLEX-LEGAL-BERT displayed almost identical results to LEGAL-BERT on the current dataset, but as the dataset would no longer contain only sentences from EU law,

we would expect that the performance by EURLEX-LEGAL-BERT would decrease and that LEGAL-BERT, which has been pretrained on far more legal data, would continue to yield similar results.

5.2.5 Improving the method for Multilingual BERT

In this work, we fine-tuned Multilingual BERT on a set of over 4000 annotated sentences from Dutch law text, to investigate whether fine-tuning BERT on more data from a different language could yield competitive results, or even better results given the larger size of the dataset. For this approach, we only fine-tuned on Dutch data and tested on English data, which in hindsight may have been a very challenging task for the model. A route that was not explored in this thesis is whether we can see an improvement in BERT's predictions if we extend the English training set with sentences from Dutch law. This approach would allow us to explore whether the model is able to leverage additional knowledge from Dutch law texts and whether this is transferable to English data. We believe this is an interesting direction for future work as it could provide a more nuanced answer to subquestion 5 and is quite easily executable given that both datasets are readily available.

5.2.6 Evaluation of the models

The chosen methods for evaluation of the predictions by our models allowed us to gain insight into the types of mistakes that the models made, and where the models differed. For example, tables 4.5 and 4.7 revealed the pattern that all models are more successful at identifying the *actor* and *action* roles, than the *object* and *recipient*. Moreover, the MUC evaluation allowed us to understand that most models especially struggled with assigning the correct span of words to the *object*, rather than the correct type of role. In short, the MUC evaluation helped us know where to look for mistakes that were made by the model. However, it does not provide a more detailed insight into these mistakes. It does not tell us whether models struggle with words at the boundaries of roles that might span multiple phrases, or if they are missing much larger chunks of the span. For this reason, we believe that a more in-depth qualitative analysis of the errors made by the models would be beneficial. Such an analysis could help us understand if the annotation protocol would need to be updated with respect to the words that should or should not be included at the start or end of roles, or if there are other issues that need to be addressed.

Another way to measure the performance of the models would be to determine how often they correctly predict the entire sentence. This will tell us not only about how well models learn certain roles but also how well they are able to capture the semantics of the entire sentence. This would be an interesting method to incorporate in future work.

5.3 Interpretation of the results

From the results, it became clear that fine-tuning language models to label Flint roles was the best approach for the task that we defined in this thesis. However, among the fine-tuned models, we observed some interesting differences. Here, we

provide some additional reflection on what may have caused these differences and what the implications may be.

We saw that M-BERT's performance was relatively close to BERT's for the *action* and *actor*. However, for the *object* and *recipient*, its scores dropped significantly. We believe that this might be a result of the differences in language use in the legal domain across languages given that legal language is often highly specific and tailored to each language's legal system. This might explain why M-BERT, which had only seen examples of Dutch legal sentences, had trouble modelling these roles. As we saw in section 4.1.2, the *object* roles were generally very long and more complex so the model may have had more trouble correctly identifying these complex roles in English, especially since it had not seen any specific examples of how these roles are usually structured in English. This gives additional emphasis to the point that was made in section 5.2.5, which highlighted the need for adding English sentences to the fine-tuning process of M-BERT.

Overall, the domain-specific models achieved the best scores and the differences between LEGAL-BERT and EURLEX-LEGAL-BERT were very small despite EURLEX-LEGAL-BERT having been pretrained on much less legal data. However, we believe that the high performance of EURLEX-LEGAL-BERT can partly be attributed to the fact that the fine-tuning dataset solely consisted of sentences from EU legislation, which is exactly what EURLEX-LEGAL-BERT was pretrained on. We expect that if we were to add sentences from other legal sources to the dataset, we would start to notice a larger difference in the results of the domain-specific models, where LEGAL-BERT would be more generalisable across the entire legal domain.

5.4 Future work

In this thesis, we developed several models that were designed to label action sentences from English legal text with the semantic roles that are associated with the first four slots of the Flint act frame. It is important to recognise that this represents just one aspect of the process of automatically filling Flint frames. We can identify many directions for future work to attain the final objective of automatic frame-filling.

The information that is currently missing to fill the Flint act frames is which *actor* performs which *action* and which *objects* and *recipients* are affected by this *action*. The current methodology stays within the bounds of the sentence and is unable to link *actors*, *objects*, and *recipients* to specific *actions*, which is essential information in the filling of frames. As such, future research could focus on developing a method that can recognise which roles are specifically tied to which actions, while considering information from longer sections of text. To achieve this, a potential solution could be to implement a more elaborate annotation process that links roles to specific actions within longer sections of legal text. Moreover, fine-tuning the Longformer architecture (Beltagy et al., 2020), a transformer architecture for longer documents, could be a suitable solution. Unlike BERT, the Longformer architecture is not constrained by a maximum input sequence length of 512 words, and, as a result, it may be better suited to capture the intricacies in longer legal texts. The downside of this approach is that executing the annotations that are required for this may become a very intricate and time-consuming task, and the task may be difficult

for the models to learn, resulting in less reliable Flint frames. A solution that is more within reach is to link the roles to the actions within sentences by annotations. Models will probably be easier to train easily to label and link within sentences than outside the sentence bounds. Subsequently, a Longformer model could be used in combination with a coreference and anaphora resolution component to link entities from other sentences in the law to the roles identified within the annotated action sentences. Whereas this type of model may not be able to capture all roles related to an *action* for an act frame, it can be used to fill Flint frames. An additional benefit of using coreference and anaphora resolution is that, after using its results to fill Flint frames, it might be also able to provide insight into whether the *actors* or *recipients* from different frames are the same entities. To determine to what extent these techniques could help the modelling of this specific dataset, it could be useful for future work to first extract some statistics from the data. For example, how often do sentences contain multiple instances of the same role, how often do sentences with one *actor* and multiple *actions* occur, and also whether the current models score lower on these sentences than on sentences that contain a maximum of one role per role type.

Other directions for future work would be to identify other elements of the Flint act frame, such as the preconditions and postconditions, or to explore how the methods from this thesis can be extended to work for the Flint fact frame.

Lastly, we highlight the use of ChatGPT as a promising avenue to explore in future work. The language model has evolved significantly over the period during which this thesis was written and presents many opportunities to automate the role labelling as well as the frame-filling process. One of the advantages of ChatGPT is that it is continually updated with new data. Consequentially, a system based on this model is more likely to remain relevant for a longer period of time as it stays up to date with the texts from the legal field. Some disadvantages associated with the use of ChatGPT are inconsistency, lack of transparency, and reliability. To fully understand the extent to which it can improve these processes, further experimentation is required.

5.5 Context and relevance

We project that there will be a significant societal relevance to a system that can automatically create Flint frames for legal text. While the system developed in this thesis focuses solely on labelling sentences from law text with semantic roles, rather than automating the entire frame-filling process, we believe there are still compelling benefits to this system. Most importantly, it provides a method that allows us to quickly understand what the *actions* in certain law texts are and who the parties involved are (*actors* and *recipients*). Whereas it does not logically structure this information in frames, it does offer an efficient insight into what information is or is not contained in a certain law. Still, automatically filling the Flint frames remains a crucial next step. Such a system would be a valuable tool to assist legal professionals in their analysis of legal text, as it can efficiently organise and extract information from these texts and it allows them to shift their focus to more high-level tasks. It should be stressed that we still consider this a supportive tool. We emphasise that it is not designed to replace legal professionals, despite the fact that it can potentially be leveraged for automated decision-making, for example on whether or

not to grant a residence permit. Legal professionals are still expected to read the law and make the decision themselves; systems that automatically fill Flint frames are only supposed to lift some of the burden of this decision-making process.

Lastly, we point out the importance of involving legal experts in the creation of Flint filling systems for it to become a standardised system that can be used in practice, on a large scale. They need to be involved in the process of designing and evaluating such a system, so that we can understand what the specific needs are and gain insight into the reliability and value of the system output.

This thesis focused on developing a model that can automatically label sentences that contain a normative action from English legal text with Flint act frame roles. To support this process, a dataset was developed with 1539 sentences from EU legislation, with annotations of these Flint roles. Seven models were developed and compared: a rule-based baseline, a mapping model, and five fine-tuned variations of BERT. After an extensive evaluation process, the fine-tuned LEGAL-BERT model was identified as the most successful approach to the task. Several directions for future work were identified to further develop the model and methodology to reach the final goal of automatically filling Flint frames.

Bibliography

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *ArXiv*, *1409*.

Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998). The Berkeley FrameNet project. *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, 86–90. https://doi.org/10.3115/980845.980860

Bakker, R., van Drie, R., de Boer, M., van Doesburg, R., & van Engers, T. (2022a). Semantic role labelling for Dutch law texts. *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 448–457. https://aclanthology.org/2022.lrec-1.47

Bakker, R., de Boer, M., van Drie, R., & Vos, D. (2022b). Extracting structured knowledge from Dutch legal texts: A rule-based approach. *EKAW - KM4LAW 2022: International Conference on Knowledge Engineering and Knowledge Management, The Knowledge Management for Law workshop.*

Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: Pretrained language model for scientific text. *EMNLP*.

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. *CoRR*, *abs/2004.05150*. https://arxiv.org/abs/2004.05150

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. *The Journal of Machine Learning Research*, *3*, 1137–1155.

Biagioli, C., Francesconi, E., Passerini, A., Montemagni, S., & Soria, C. (2005). Automatic semantics extraction in law documents. *Proceedings of the 10th International Conference on Artificial Intelligence and Law*, 133–140. https://doi.org/10.1145/1165485.1165506

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. https://doi.org/10.1162/tacl_a_00051

Bozinovski, S., & Fulgosi, A. (1976). The influence of pattern similarity and transfer learning upon training of a base perceptron b2. *Proceedings of Symposium Informatica*, *3*, 121–126.

Breaux, T. (2009). *Legal requirements acquisition for the specification of legally compliant information systems* (Doctoral dissertation) [AAI3357689]. North Carolina State University.

Brighi, R., Lesmo, L., Mazzei, A., Palmirani, M., & Radicioni, D. P. (2008). Towards semantic interpretation of legal modifications through deep syntactic analysis. In E. Francesconi, G. Sartor, & D. Tiscornia (Eds.), *Legal knowledge and information systems - JURIX 2008: The twenty-first annual conference on legal knowledge and information systems, florence, italy, 10-13 december 2008* (pp. 202–206). IOS Press. https://doi.org/10.3233/978-1-58603-952-3-202

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., ... Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

Chalkidis, I., & Kampas, D. (2019). Deep learning in law: Early adaptation and legal word embeddings trained on large corpora. *Artificial Intelligence and Law*, *27*, 171–198. https://doi.org/https://doi.org/10.1007/s10506-018-9238-9

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., & Androutsopoulos, I. (2020). LEGAL-BERT: The muppets straight out of law school. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2898–2904. https://doi.org/10.18653/v1/2020.findings-emnlp.261

Chinchor, N., & Sundheim, B. (1993). MUC-5 evaluation metrics. *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*. https://aclanthology.org/M93-1007

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734. https://doi.org/10.3115/v1/D14-1179

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

Di Fabio, A., Conia, S., & Navigli, R. (2019). VerbAtlas: A novel large-scale verbal semantic resource and its application to semantic role labeling. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 627–637. https://doi.org/10.18653/v1/D19-1058

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, *26*(3), 297–302. http://www.jstor.org/stable/1932409

van Doesburg, R. (2017). *A formal method for interpretation of sources of norms* (tech. rep.). Leibniz Center for Law, University of Amsterdam.

van Doesburg, R., & van Engers, T. (2019). Explicit interpretation of the Dutch Aliens Act. *CEUR Workshop Proceedings, 1st Workshop on Artificial Intelligence and the Administrative State, AIAS 2019*, 27–37.

van Drie, R., de Boer, M., Bakker, R., Tolios, I., & Vos, D. (2023). The Dutch law as a semantic role labelling dataset [in press]. *Nineteenth International Conference on Artificial Intelligence and Law*.

Elwany, E., Moore, D., & Oberoi, G. (2019). BERT goes to law school: Quantifying the competitive advantage of access to large legal corpora in contract understanding. *Workshop on Document Intelligence at NeurIPS 2019*. https://openreview.net/forum?id=rkeRMT9cLH

Fellbaum, C. (Ed.). (1998). *WordNet: An electronic lexical database*. MIT Press.

Fillmore, C. J. (1968). The case for case, dins. In E. Bach & R. Harms (Eds.), *Universals in linguistic theory*. Holt, Rinehart, Winston.

Fleiss, J. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, *76*(5), 378–382. https://doi.org/https://doi.org/10.1037/h0031619

Gao, X., & Singh, M. (2014). Extracting normative relationships from business contracts. *13th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2014*, *1*, 101–108.

Gao, X., & Ichise, R. (2017). Adjusting word embeddings by deep neural networks. *Proceedings of the 9th International Conference on Agents and Artificial Intelligence (ICAART 2017)International Conference on Agents and Artificial Intelligence*, 398–406.

Gildea, D., & Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational Linguistics*, *28*(3), 245–288. https://doi.org/10.1162/089120102760275983

Gruber, J. (1965). *Studies in lexical relations* (Ph.D. Thesis). Massachussets Institute of Technology.

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., & Poon, H. (2021). Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthcare*, *3*(1). https://doi.org/10.1145/3458754

He, L., Lee, K., Lewis, M., & Zettlemoyer, L. (2017). Deep semantic role labeling: What works and what's next. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 473–483. https://doi.org/10.18653/v1/P17-1044

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*, 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735

Hohfeld, W. (1917). Fundamental legal conceptions as applied in judicial reasoning. *The Yale Law Journal*, *26*(8), 710–770. http://www.jstor.org/stable/786270

Howard, J., & Ruder, S. (2018). Fine-tuned language models for text classification. *CoRR*, *abs/1801.06146*. http://arxiv.org/abs/1801.06146

Huang, E., Socher, R., Manning, C., & Ng, A. (2012). Improving word representations via global context and multiple word prototypes. *50th Annual Meeting of the Association for Computational Linguistics, ACL 2012 - Proceedings of the Conference*, *1*, 873–882.

Huang, K., Altosaar, J., & Ranganath, R. (2019). ClinicalBERT: modeling clinical notes and predicting hospital readmission. *CoRR*, *abs/1904.05342*. http://arxiv.org/abs/1904.05342

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., & Levy, O. (2020). SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, *8*, 64–77. https://doi.org/10.1162/tacl_a_00300

Jurafsky, D., & Martin, J. (2021). Speech and language processing. Stanford University.

van Kralingen, R. (1995). *Frame-based conceptual models of statute law* (Doctoral dissertation). Netherlands, Kluwer Law International.

Labutov, I., & Lipson, H. (2013). Re-embedding words. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 489–493. https://aclanthology.org/P13-2087

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 159–174.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2019). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, *36*(4), 1234–1240. https://doi.org/10.1093/bioinformatics/btz682

de Marneffe, M.-C., Manning, C., Nivre, J., & Zeman, D. (2021). Universal Dependencies. *Computational Linguistics*, *47*(2), 255–308. https://doi.org/10.1162/coli_a_00402

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. http://arxiv.org/abs/1301.3781

Minsky, M. (1974). *A framework for representing knowledge* (tech. rep.). USA, Massachusetts Institute of Technology.

Mitsumori, T., Murata, M., Fukuda, Y., Doi, K., & Doi, H. (2005). Semantic role labeling using support vector machines. *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, 197–200. https://aclanthology.org/W05-0629

Mnih, A., & Hinton, G. (2008). A scalable hierarchical distributed language model. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2008/file/1e056d2b0ebd5c878c550da6ac5d3724-Paper.pdf

Nazarenko, A., Lévy, F., & Wyner, A. (2022). Towards a methodology for formalising legal texts: A comparison of two experiments. *Proceedings of the International Workshop on Methodologies for Translating Legal Norms into Formal Representations (LN2FR 2022)*, 36–49.

OpenAI. (2023). Gpt-4 technical report.

Ouchi, H., Shindo, H., & Matsumoto, Y. (2018). A span selection model for semantic role labeling. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 1630–1642. https://doi.org/10.18653/v1/D18-1191

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, *31*(1), 71–106. https://doi.org/10.1162/0891201053630264

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: global vectors for word representation. *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. http://www.aclweb.org/anthology/D14-1162

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237. https://doi.org/10.18653/v1/N18-1202

Pires, T., Schlinger, E., & Garrette, D. (2019). How multilingual is Multilingual BERT? *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4996–5001. https://doi.org/10.18653/v1/P19-1493

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., & Zhang, Y. (2012). CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes.

Joint Conference on EMNLP and CoNLL - Shared Task, 1–40. https://aclanthology.org/W12-4501

Radford, A., & Narasimhan, K. (2018). *Improving language understanding by generative pre-training* (tech. rep.).

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language models are unsupervised multitask learners* (tech. rep.).

Schuler, K. K., & Palmer, M. (2005). VerbNet: A broad-coverage, comprehensive verb lexicon.

Segura-Bedmar, I., Martínez, P., & Herrero-Zazo, M. (2013). SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, 341–350. https://aclanthology.org/S13-2056

Shaghaghian, S., Feng, L., Jafarpour, B., & Pogrebnyakov, N. (2020). Customizing contextualized language models for legal document reviews. *2020 IEEE International Conference on Big Data (Big Data)*, 2139–2148.

Shi, P., & Lin, J. (2019). Simple BERT models for relation extraction and semantic role labeling. *CoRR, abs/1904.05255*. http://arxiv.org/abs/1904.05255

Singh, M. (2014). Norms as a basis for governing sociotechnical systems. *ACM Trans. Intell. Syst. Technol.*, *5*(1), 4207–4211. https://doi.org/10.1145/2542182.2542203

Uebersax, J. S. (1982). A design-independent method for measuring the reliability of psychiatric diagnosis. *Journal of Psychiatric Research*, *17*(4), 335–342. https://doi.org/https://doi.org/10.1016/0022-3956(82)90039-5

Uyttendaele, C., Moens, M.-F., & Dumortier, J. (1998). Salomon: Automatic abstracting of legal cases for effective access to court decisions. *Artificial Intelligence and Law*, *6*, 59–79.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010.

de Vries, W., van Cranenburgh, A., Bisazza, A., Caselli, T., van Noord, G., & Nissim, M. (2019). BERTje: A dutch BERT model. *CoRR, abs/1912.09582*. http://arxiv.org/abs/1912.09582

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 353–355. https://doi.org/10.18653/v1/W18-5446

Wang, W., Bi, B., Yan, M., Wu, C., Bao, Z., Xia, J., Peng, L., & Si, L. (2019). StructBERT: Incorporating language structures into pre-training for deep language understanding. *arXiv preprint arXiv:1908.04577*.

Yang, Y., Uy, M. S., & Huang, A. (2020). FinBERT: A pretrained language model for financial communications. *Contemporary Accounting Research*, *40*(2), 806–841. https://doi.org/10.1111/1911-3846.12832

Zhang, H., Ro, J., & Sproat, R. (2020). Semi-supervised URL segmentation with recurrent neural networks pre-trained on knowledge graph entities. *The 28th International Conference on Computational Linguistics (COLING 2020)*.

Zheng, Z., Lu, X.-Z., Chen, K.-Y., Zhou, Y.-C., & Lin, J.-R. (2022). Pretrained domain-specific language model for natural language processing tasks in the AEC domain. *Computers in Industry*, *142*, 103733. https://doi.org/10.1016/j.compind.2022.103733

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *The IEEE International Conference on Computer Vision (ICCV)*.

A Appendix

A.1 Example of a Flint Act Frame

Table A.1 shows an example of a manually created Flint act frame, which is based on Art. 5 of the GDPR. The article itself is presented below. Within the article, we have highlighted the text corresponding to how we would annotate the sentences in the experiment. Actions have been marked green, actors blue, objects red, and recipients yellow. In table A.1 we can see that these markings do not directly correspond to what has been manually filled into the slots in the act frame. This demonstrates how the identification of Flint act roles is not yet sufficient to build the Flint frames with. In this example, there are Moreover, the frame in the example is also based on expert knowledge.

Art. 5 GDPR[1]

Principles relating to processing of personal data

1. Personal data shall be:

   (a) processed lawfully, fairly and in a transparent manner in relation to the data subject (‘lawfulness, fairness and transparency’);

   (b) collected for specified, explicit and legitimate purposes and not further processed in a manner that is incompatible with those purposes; further processing for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes shall, in accordance with Article 89(1), not be considered to be incompatible with the initial purposes (‘purpose limitation’);

   (c) adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed (‘data minimisation’);

   (d) accurate and, where necessary, kept up to date; every reasonable step must be taken to ensure that personal data that are inaccurate, having regard to the purposes for which they are processed, are erased or rectified without delay (‘accuracy’);

   (e) kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed; personal data may be stored for longer periods insofar as the personal data will be processed solely for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes in accordance with Article 89(1) subject to implementation of the appropriate technical and organisational measures required by this Regulation in order to safeguard the rights and freedoms of the data subject (‘storage limitation’);

   (f) processed in a manner that ensures appropriate security of the personal data, including protection against unauthorised or unlawful processing and against accidental loss, destruction or damage, using appropriate technical or organisational measures (‘integrity and confidentiality’).

2. The controller shall be responsible for, and able to demonstrate compliance with, paragraph 1 (‘accountability’).

[1] https://gdpr-info.eu/art-5-gdpr/

Act	<<collect personal data>>
Action	collect
Actor	[processor]
Object	[personal data]
Recipient	[data subject]
Precondition	<personal data are processed lawfully, fairly and in a transparent manner in relation to the data subject> AND <personal data are collected for specified, explicit and legitimate purposes> AND NOT <personal data are further processed in a manner that is incompatible with the purposes for which they were collected> AND <personal data are adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed> AND <personal data are accurate and kept up to date> AND <personal data are kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed> AND <personal data are processed in a manner that ensures appropriate security of the personal data>
Creating postcondition	<controller shall be able to demonstrate compliance with Art. 5(1) GDPR>
Terminating postcondition	-
Source	Art. 5(1) GDPR

Table A.1: Example of manually created act frame from the GDPR, where <<>> is used to denote an act, <> to denote a duty and [] to denote a fact. Taken from Bakker et al. (2022a)

A.2 Annotation protocol

Task

You will be presented with sentences from law text. In these sentences, you are supposed to select words with the following roles/functions:

- Action: that what happens (often a verb)
- Actor: the volitional causer of the event
- Object: the entity which is moved by the action / the entity undergoing the effect of the action
- Recipient: the entity for whose benefit the action was performed

We can assign these roles by asking: who (actor) does (action) what (object) to whom (recipient)?

For each sentence, try to indicate as completely as possible which words have these roles. It is possible that a sentence does not contain an object, actor, or recipient. In this case, it is not necessary to select that role. It is also possible that there are multiple words in the sentence that have the same type of role. In this case, they should all be selected as such.

Extra information

Below you will find an overview of the different types of words or phrases that should or should not be included in the annotations. While annotating, you can return to this overview at any given time.

Include in the annotations:

- Articles (lidwoorden) should be included in the actor, object and recipient.
  Example: [**The controller**]$_{\text{ACTOR}}$ [shall provide]$_{\text{ACTION}}$ [information]$_{\text{OBJECT}}$.
- Prepositions should be included in the actor, object and recipient.
  Example 1: [The Member State]$_{\text{ACTOR}}$ [makes]$_{\text{ACTION}}$ [the information]$_{\text{OBJECT}}$ [available]$_{\text{ACTION}}$ [**to** the data subject]$_{\text{RECIPIENT}}$.
  Example 2: [The controller]$_{\text{ACTOR}}$ [asks]$_{\text{ACTION}}$ [**to** be included in all communications]$_{\text{OBJECT}}$ and [informs]$_{\text{ACTION}}$ [the data subjects]$_{\text{RECIPIENT}}$ [**of** their rights]$_{\text{OBJECT}}$.
- Complementisers should be included in the object, actor and recipient (a complementiser is a conjunction that can be used at the start of a clause, which allows the entire clause to function as the object of the sentence).
  Example: [The supervisory authority]$_{\text{ACTOR}}$ [determines]$_{\text{ACTION}}$ [**that** the processing was not lawful]$_{\text{OBJECT}}$.
- Negations should be included in the action.
  Example: [The Member State]$_{\text{ACTOR}}$ [shall **not** provide]$_{\text{ACTION}}$ [the information referred to in paragraph 5]$_{\text{OBJECT}}$.

- Multiple instances of the same role. If a sentence contains multiple actors, actions, objects or recipients they should all be annotated as such.
  Example: [The supervisory authority]$_{\text{ACTOR}}$ [decides]$_{\text{ACTION}}$ [on the case]$_{\text{OBJECT}}$ and [informs]$_{\text{ACTION}}$ [the Member State]$_{\text{RECIPIENT}}$ [of the decision]$_{\text{OBJECT}}$.

- Phrasal verbs should be included in the action. The adverb of preposition of a phrasal verb should be included in your annotation of the action.
  Example: [The board]$_{\text{ACTOR}}$ [calls **off**]$_{\text{ACTION}}$ [the meeting]$_{\text{OBJECT}}$ in case of any cancellations.

- Words that are essential to the meaning of the action should be included in the action.
  Example 1: [The authorities]$_{\text{ACTOR}}$ [take]$_{\text{ACTION}}$ the necessary [**steps**]$_{\text{ACTION}}$ to enforce the rules.
  Example 2; [The Union]$_{\text{ACTOR}}$ [shall make **public**]$_{\text{ACTION}}$ [all relevant communications]$_{\text{OBJECT}}$.

- Include interpunction in the annotation only if it appears within the role. Do not include periods or commas that appear at the beginning or end of the role.

Do not include in the annotations:

- Adverbs (something that modifies the verb) should not be included in the action.
  Example: [The supervisory authority]$_{\text{ACTOR}}$ **immediately** [informs]$_{\text{ACTION}}$ [the board]$_{\text{RECIPIENT}}$.

- Certain clauses (e.g. preconditions) should not be included in the annotation, even if the information contained in them is important or essential for the meaning of the sentence. We are referring to the type of clause that is a word or a phrase that can be omitted without making the sentence grammatically incorrect.
  Example 1: **Where the data subject agrees**, [the controller]$_{\text{ACTOR}}$ [shall share]$_{\text{ACTION}}$ [the data]$_{\text{OBJECT}}$ [with a third party]$_{\text{RECIPIENT}}$.
  Example 2: [The board]$_{\text{ACTOR}}$ [takes responsibility]$_{\text{ACTION}}$ [for this decision]$_{\text{OBJECT}}$, **unless otherwise provided for in this regulation**.
  Example 3: [The board]$_{\text{ACTOR}}$ [shall define]$_{\text{ACTION}}$ [the division of tasks]$_{\text{OBJECT}}$ **in the first chapter of their regulation**.

- Clusters of verbs should not be included in the action as a whole. In many cases, the cluster of verbs is not all part of the action, but should be split up to form part of the action and part of the object.
  Example: [The Commission]$_{\text{ACTOR}}$ [**may**]$_{\text{ACTION}}$ ultimately [**decide**]$_{\text{ACTION}}$ [**to handle** the case]$_{\text{OBJECT}}$.

- Actions (and their corresponding actors, objects and recipients) that are part of clauses should not be included in the annotation.
  Example: [**When the supervisory authority defines a transfer as lawful**]$_{\text{PRECONDITION}}$, [the processor]$_{\text{ACTOR}}$ [may execute]$_{\text{ACTION}}$ [the transfer]$_{\text{OBJECT}}$.

Please also pay attention to the following:

- Passive sentences. For sentences written in the passive voice, it is important to consider who/what performs the action and who/what undergoes the action. In passive sentences, the grammatical subject might be the thing acted upon, rather than the actor.
  Example 1: [The information]$_{\text{OBJECT}}$ [shall be made public]$_{\text{ACTION}}$.
  Example 2: [The supervisory authority]$_{\text{RECIPIENT}}$ [shall be assisted]$_{\text{ACTION}}$ [by a committee]$_{\text{ACTOR}}$.
  Example 3: [This power]$_{\text{OBJECT}}$ [may be assigned]$_{\text{ACTION}}$ [to the Member State]$_{\text{RECIPIENT}}$ [by the supervisory authority]$_{\text{ACTOR}}$.

A.3 Hyperparameter optimisation results

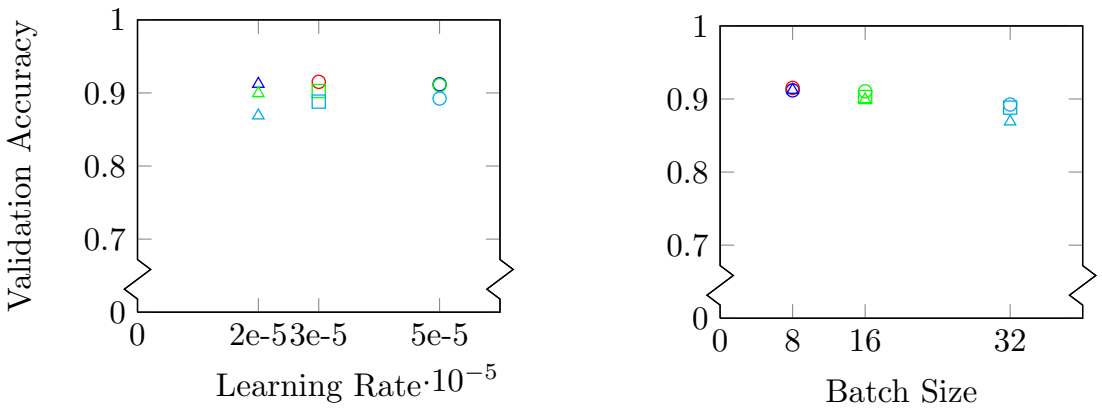

Figure A.1: BERT hyperparameter optimisation results

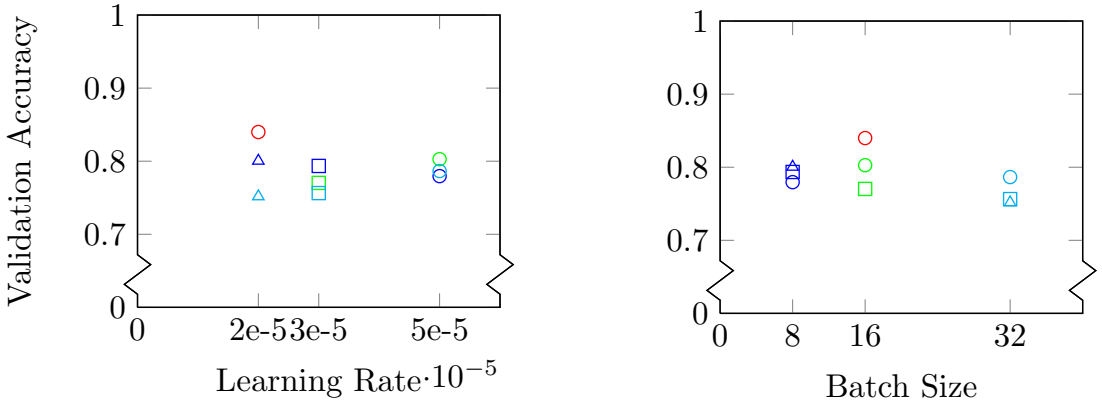

Figure A.2: M-BERT hyperparameter optimisation results

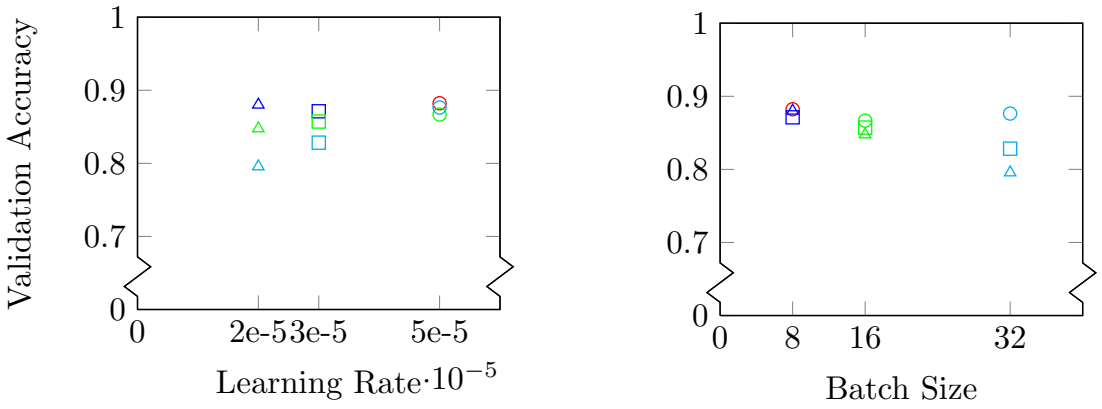

Figure A.3: SpanBERT hyperparameter optimisation results

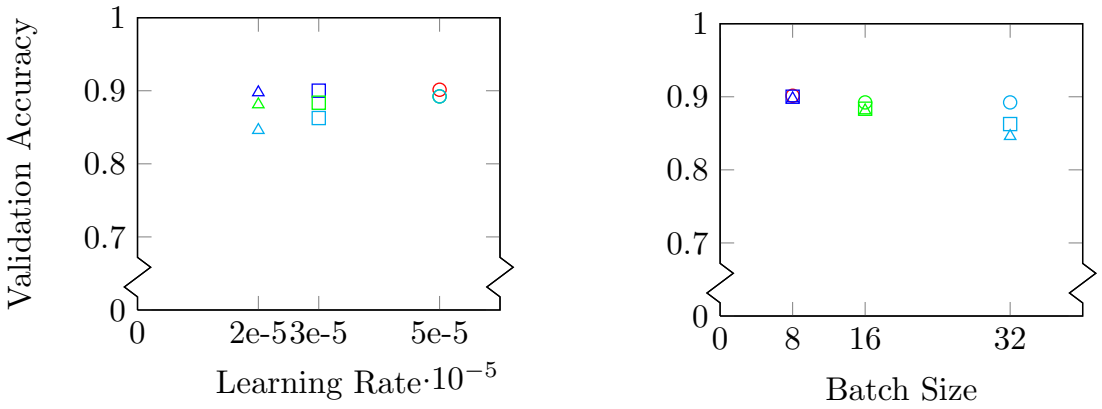

Figure A.4: LEGAL-BERT hyperparameter optimisation results

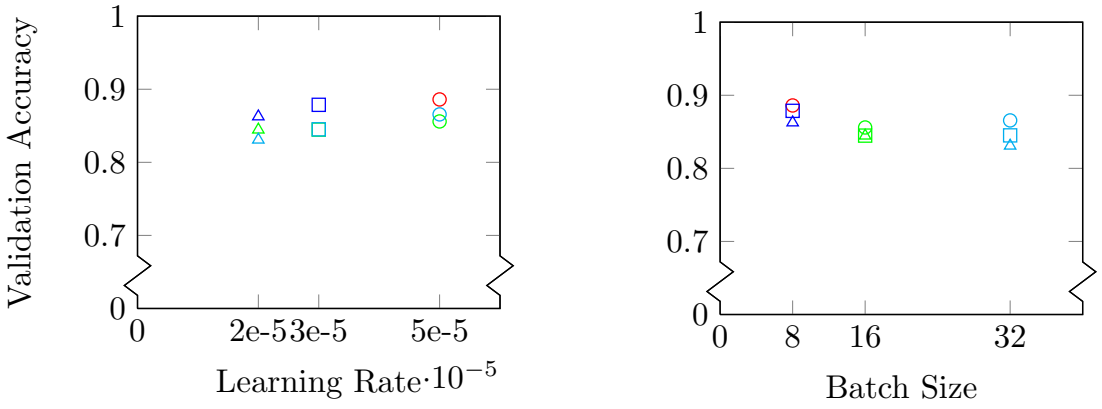

Figure A.5: EURLEX-LEGAL-BERT hyperparameter optimisation results

A.4 Confusion matrices

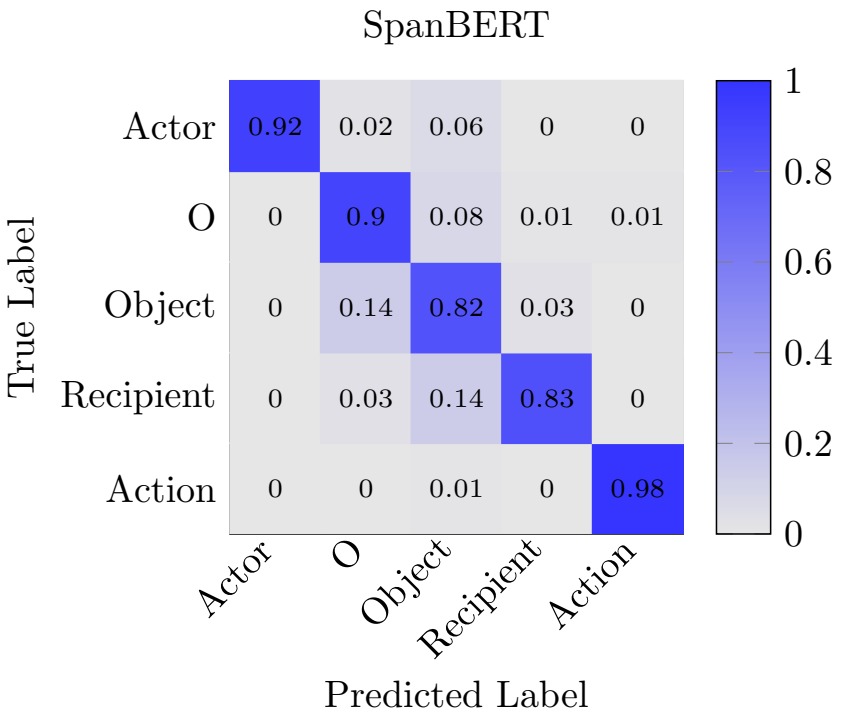

Figure A.6: Mean confusion matrix for SpanBERT

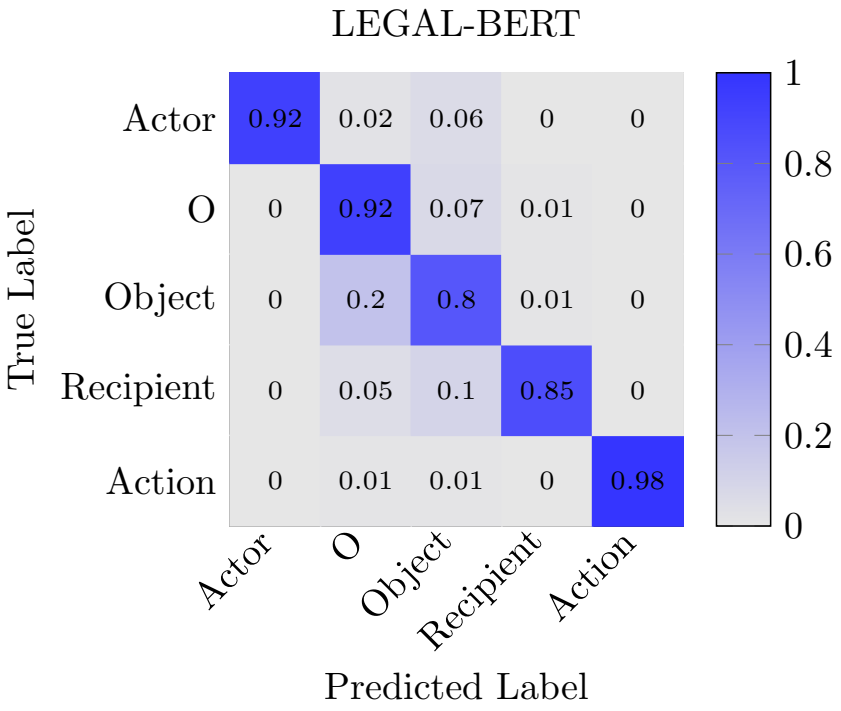

Figure A.7: Mean confusion matrix for LEGAL-BERT

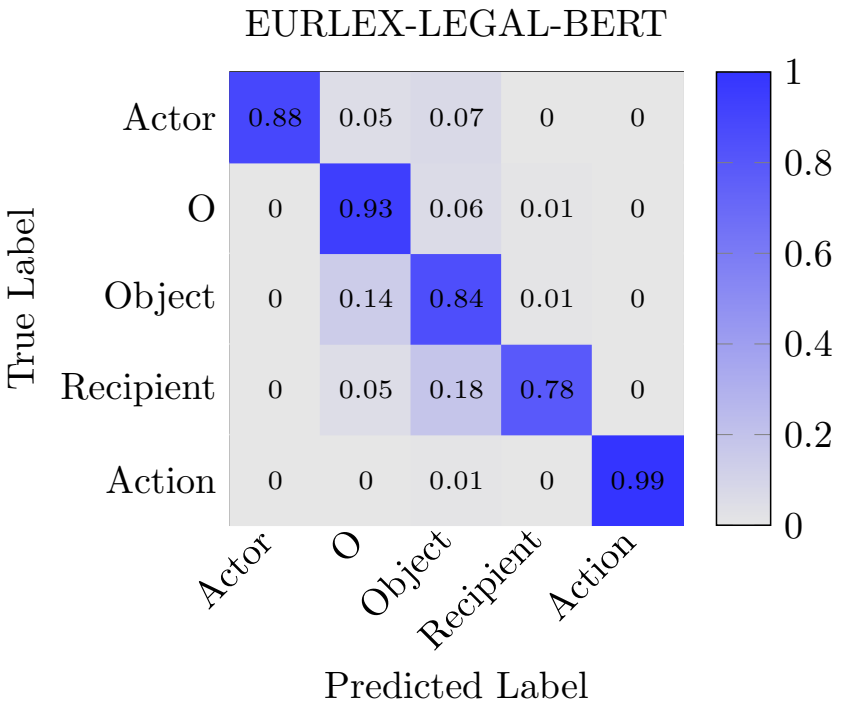

Figure A.8: Mean confusion matrix for EURLEX-LEGAL-BERT

A.5 MUC evaluation results

A.5.1 Overall results

Table A.2: Overall MUC results for the rule-based model

Measure	Type	Partial	Exact	Strict
Correct	391	253	253	209
Incorrect	42	0	180	114
Partial	0	180	0	0
Missed	107	107	107	107
Spurious	328	328	328	328
F1	0.601	0.527	0.389	0.375

Table A.3: Overall MUC results for the mapping model

Measure	Type	Partial	Exact	Strict
Correct	382	242	242	230
Incorrect	26	0	166	178
Partial	0	166	0	0
Missed	132	132	132	132
Spurious	21	21	21	21
F1	0.788	0.671	0.499	0.475

Table A.4: Overall MUC results for the fine-tuned BERT model

Measure	Type	Partial	Exact	Strict
Correct	506	448	448	441
Incorrect	15	0	73	80
Partial	0	73	0	0
Missed	19	19	19	19
Spurious	56	56	56	56
F1	0.906	0.866	0.802	0.790

Table A.5: Overall MUC results for the fine-tuned Multilingual BERT model

Measure	Type	Partial	Exact	Strict
Correct	483	385	385	376
Incorrect	28	0	126	135
Partial	0	126	0	0
Missed	29	29	29	29
Spurious	173	173	173	173
F1	0.789	0.732	0.629	0.614

Table A.6: Overall MUC results for the fine-tuned SpanBERT model

Measure	Type	Partial	Exact	Strict
Correct	502	448	448	440
Incorrect	13	0	67	75
Partial	0	67	0	0
Missed	25	25	25	25
Spurious	48	48	48	48
F1	0.910	0.873	0.812	0.798

Table A.7: Overall MUC results for the fine-tuned LEGAL-BERT model

Measure	Type	Partial	Exact	Strict
Correct	507	457	457	451
Incorrect	10	0	60	66
Partial	0	60	0	0
Missed	23	23	23	23
Spurious	40	40	40	40
F1	0.924	0.888	0.833	0.822

Table A.8: Overall MUC results for the fine-tuned EURLEX-LEGAL-BERT model

Measure	Type	Partial	Exact	Strict
Correct	503	455	455	447
Incorrect	13	0	61	69
Partial	0	61	0	0
Missed	24	24	24	24
Spurious	41	41	41	41
F1	0.917	0.885	0.830	0.815

A.5.2 Results per role

Table A.9: MUC results per role for the rule-based model

	Actor				**Action**				Object				Recipient			
Measure	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict
Correct	134	103	103	100	142	109	109	109	114	34	34	34	1	7	7	1
Incorrect	4	0	35	38	7	0	40	40	18	0	98	98	13	0	7	13
Partial	0	35	0	0	0	40	0	0	0	98	0	0	0	7	0	0
Missed	5	5	5	5	52	52	52	52	16	16	16	16	34	34	34	34
Spurious	89	89	89	89	72	72	72	72	165	165	165	165	2	2	2	2
F1	0.72	0.65	0.56	0.54	0.67	0.61	0.52	0.52	0.51	0.37	0.15	0.15	0.03	0.33	0.22	0.03

Table A.10: MUC results per role for the mapping model

	Actor				**Action**				**Object**				**Recipient**			
Measure	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict
Correct	121	117	117	113	135	16	16	16	102	85	85	81	24	24	24	20
Incorrect	7	0	11	15	4	0	123	123	8	0	25	29	7	0	7	11
Partial	0	11	0	0	0	123	0	0	0	25	0	0	0	7	0	0
Missed	15	15	15	15	62	62	62	62	38	38	38	38	17	17	17	17
Spurious	1	1	1	1	7	7	7	7	11	11	11	11	2	2	2	2
F1	0.89	0.90	0.86	0.83	0.78	0.45	0.09	0.09	0.76	0.72	0.63	0.60	0.59	0.68	0.59	0.49

Table A.11: MUC results per role for the fine-tuned BERT model

	Actor				**Action**				**Object**				**Recipient**			
Measure	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict
Correct	138	134	134	131	191	179	179	178	137	103	103	101	40	32	32	31
Incorrect	3	0	7	10	3	0	15	16	6	0	40	42	3	0	11	12
Partial	0	7	0	0	0	15	0	0	0	40	0	0	0	11	0	0
Missed	2	2	2	2	7	7	7	7	5	5	5	5	5	5	5	5
Spurious	0	0	0	0	6	6	6	6	35	35	35	35	15	15	15	15
F1	0.97	0.97	0.94	0.92	0.95	0.93	0.89	0.89	0.84	0.75	0.63	0.62	0.75	0.71	0.60	0.58

Table A.12: MUC results per role for the fine-tuned Multilingual BERT model

	Actor				Action				Object				Recipient			
Measure	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict
Correct	137	127	127	122	181	151	151	151	130	80	80	78	35	27	27	25
Incorrect	5	0	15	20	4	0	34	34	14	0	64	66	5	0	13	15
Partial	0	15	0	0	0	34	0	0	0	64	0	0	0	13	0	0
Missed	1	1	1	1	16	16	16	16	4	4	4	4	8	8	8	8
Spurious	20	20	20	20	13	13	13	13	126	126	126	126	14	14	14	14
F1	0.90	0.88	0.83	0.80	0.91	0.84	0.76	0.76	0.62	0.53	0.38	0.37	0.69	0.66	0.53	0.49

Table A.13: MUC results per role for the fine-tuned SpanBERT model

	Actor				Action				Object				Recipient			
Measure	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict
Correct	137	136	136	133	187	167	167	167	136	110	110	107	42	35	35	33
Incorrect	4	0	5	8	1	0	21	21	6	0	32	35	2	0	9	11
Partial	0	5	0	0	0	21	0	0	0	32	0	0	0	9	0	0
Missed	2	2	2	2	13	13	13	13	6	6	6	6	4	4	4	4
Spurious	1	1	1	1	8	8	8	8	32	32	32	32	7	7	7	7
F1	0.96	0.97	0.95	0.93	0.94	0.89	0.84	0.84	0.85	0.78	0.68	0.66	0.85	0.80	0.71	0.67

Table A.14: MUC results per role for the fine-tuned LEGAL-BERT model

	Actor				Action				Object				Recipient			
Measure	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict
Correct	137	137	137	133	194	180	180	180	135	106	106	104	41	34	34	34
Incorrect	4	0	4	8	1	0	15	15	5	0	34	36	0	0	7	7
Partial	0	4	0	0	0	15	0	0	0	34	0	0	0	7	0	0
Missed	2	2	2	2	6	6	6	6	8	8	8	8	7	7	7	7
Spurious	0	0	0	0	4	4	4	4	29	29	29	29	7	7	7	7
F1	0.96	0.98	0.96	0.94	0.97	0.94	0.90	0.90	0.85	0.78	0.67	0.66	0.85	0.78	0.71	0.71

Table A.15: MUC results per role for the fine-tuned EURLEX-LEGAL-BERT model

	Actor				Action				Object				Recipient			
Measure	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict	Type	Partial	Exact	Strict
Correct	136	134	134	130	194	184	184	184	134	104	104	102	39	33	33	31
Incorrect	4	0	6	10	1	0	11	11	6	0	36	38	2	0	8	10
Partial	0	6	0	0	0	11	0	0	0	36	0	0	0	8	0	0
Missed	3	3	3	3	6	6	6	6	8	8	8	8	7	7	7	7
Spurious	2	2	2	2	5	5	5	5	28	28	28	28	6	6	6	6
F1	0.95	0.96	0.94	0.91	0.97	0.94	0.92	0.92	0.85	0.77	0.66	0.65	0.82	0.78	0.69	0.65