Semantic role extraction in law texts: a comparative analysis of language models for legal information extraction

Bakker, R.M.; Schoevers, A.J.; Drie, R.A.N. van; Schraagen, M.P.; Boer, M.H.T. de

Artificial Intelligence and Law
https://doi.org/10.1007/s10506-025-09437-x

ORIGINAL RESEARCH

Semantic role extraction in law texts: a comparative analysis of language models for legal information extraction

Roos M. Bakker[1,2] · **Akke J. Schoevers**[1,3] · **Romy A. N. van Drie**[1] · **Marijn P. Schraagen**[3] · **Maaike H. T. de Boer**[1]

Accepted: 27 January 2025

Abstract

Norms are essential in our society: they dictate how individuals should behave and interact within a community. They can be written down in laws or other written sources. Interpretations often differ; this is where formalisations offer a solution. They express an interpretation of a source of norms in a transparent manner. However, creating these interpretations is labour intensive. Natural language processing techniques can support this process. Previous work showed the potential of transformer-based models for Dutch law texts. In this paper, we (1) introduce a dataset of 2335 English sentences annotated with legal semantic roles conform the Flint framework; (2) fine-tune a collection of language models on this dataset, and (3) query two non-fine-tuned generative large language models (LLMs). This allows us to compare performance of fine-tuned domain-specific, task-specific, and general language models with non-fine-tuned generative LLMs. The results show that models fine-tuned on our dataset have the best performance (accuracy around 0.88). Furthermore, domain-specific models perform better than general models, indicating that domain knowledge is of added value for this task. Finally, different methods of querying LLMs perform unsatisfactorily, with maximum accuracy scores around 0.6. This indicates that for specific tasks, such as this adaptation of semantic role labelling, the process of annotating data and fine-tuning a smaller language model is preferred over querying a generative LLM, especially when domain-specific models are available.

Keywords Semantic role labelling · Large language models · Legislation · Legal information extraction · Legal semantic roles

✉ Roos M. Bakker
roos.bakker@tno.nl

1 TNO, Department Data Science, The Hague, The Netherlands

2 University of Leiden, Centre for Linguistics, Leiden, The Netherlands

3 University of Utrecht, Natural Language Processing, Utrecht, The Netherlands

Published online: 24 March 2025

1 Introduction

Norms play a crucial role in shaping and maintaining a functional and organised society. Norms are social expectations and guidelines that dictate how individuals should behave, interact, and conform within a given community. Some norms are captured in laws. Other written sources of norms include agreements, contracts and policy guidelines. Researchers have worked on formalising norms into machine-readable formats since the 1990s (Leone 2021; van Doesburg and van Engers 2019). Such formalisations can help to resolve conflicting interpretations of sources of norms, and ultimately resolve legal conflicts.

Flint is a formal language that captures an interpretation of a norm (van Doesburg and van Engers 2019). It can be used as a basis for a reasoner to model agent behaviour in a normative setting or for conflict resolution later in the process. It represents norms in terms of normative acts — who may do what to whom — and the accompanying pre- and post-conditions — when can these acts be performed, and what are the results of performing it. The normative act, with its components actor, action, object and recipient, and their pre- and post-conditions, is formalised in a frame, called a Flint frame. An example of a Flint act frame is given in Table 1.

A limitation of implementing the Flint framework or similar formalisations in real-life settings is that constructing the frames requires a lot of manual labour and domain knowledge. Parts of these frames are, however, repetitive or could be automatically deducted using linguistic patterns. Therefore, we explore a system that can support the norm modeller in their endeavour. More specifically, we explore several methods to identify the following components of an act frame: the action, actor, object and recipient. These roles are similar to semantic, or thematic roles, making the task of identifying them akin to that of semantic role labelling. We have seen several attempts to employ NLP techniques to formalise the interpretations of norms (Biagioli et al. 2005; Brighi et al. 2008). Earlier work focusing on Dutch law texts has shown the feasibility of extracting semantic roles using transformers (Bakker et al. 2022a; van Drie et al. 2023).

Table 1 Simplified example of a manually created act frame from the GDPR, adapted from Bakker et al. (2022a)

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
Creating postcondition	Controller shall be able to demonstrate compliance with Art. 5(1) GDPR
Terminating postcondition	–
Source	Art. 5 (1) GDPR

With the introduction of the transformer architecture (Vaswani et al. 2017) and subsequent language models such as BERT (Devlin et al. 2018), language models achieved enhanced performance on a variety of tasks and became widely accessible. Domain-specific models further pushed the state-of-the-art for tasks within specialised domains such as the medical field. For the legal domain, previous work showed the potential of fine-tuned language models for Dutch (Bakker et al. 2022a; van Drie et al. 2023). Recent work on generative Large Language Models (LLMs) shows their potential on a variety of tasks. Models such as Generative Pre-Trained Transformers (GPT) follow a decoder-only architecture, thus allowing for more efficient training (Brown et al. 2020). Being a thousand times larger than previous models such as BERT (Devlin et al. 2018), these models have gained popularity due to their emergent behaviour and superior performance on benchmark tasks (Wei et al. 2022). With the introduction of ChatGPT,[1] LLMs became accessible to the wider public. However, its performance on specialised tasks often shows lack of transparency and inconsistency (Schwartz et al. 2023).

The contributions of this paper are (1) a dataset of sentences from legal texts annotated with Flint roles, (2) a collection of fine-tuned models for labelling legal texts, and (3) an extensive comparison between fine-tuned domain-specific models, task-specific models, and pre-trained, non fine-tuned, generative large language models (GPT-3.5 Turbo and GPT-4). Where possible, we give results for English and Dutch. By creating a dataset for English, we can fine-tune domain- and task-specific models such as LEGAL-BERT by Chalkidis et al. (2020) and SpanBERT by Joshi et al. (2020) that do not exist for Dutch. By comparing such models to standard BERT models, we can investigate the influence of domain knowledge. For comparing the performance of fine-tuned models to generative large language models, (GPT-3.5 Turbo and GPT-4) we use four different query methods, but we do not fine-tune the LLMs. This is done to explore the potential of these generative LLMs on performing a task without the extensive effort of annotating a dataset. We compare their performance against the performance of the task-specific fine-tuned BERT models for Dutch and English.

In the next section, we describe related work on information extraction in NLP, specifically BERT and GPT, NLP for legal text and formalising norms. In Sect. 3, we describe the collection and annotation of our English dataset, all models that we tested and compared, including a rule-based approach, five models that we fine-tune on our dataset, and a mapping model which maps traditional semantic roles to Flint roles, and lastly the different approaches we used for querying two generative LLMs. In Sect. 4, we show and describe the performance of all different models and approaches on labelling legal texts with Flint roles. These results will be further discussed and interpreted in Sect. 5. Finally, we summarise our work and reflect on it in Sect. 6.

[1] https://chat.openai.com/chat

2 Related work

2.1 Formalising norms: flint

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 formalised the relations between two adversarial parties in law cases and differentiated four legal concepts which exist exclusively in pairs: Power–Liability, where a party is allowed by norms to perform an action that has effect on another party, and Duty–Claim, where a party is expected to do something for the benefit for the other party. This work can be seen as a basis to conceptualise the legal relations between two parties (van Doesburg 2017).

The Flint (Formal Language for the Interpretation of Normative Theories) language, proposed by van Doesburg (2017), is based on Hohfeld's approach to norm representation. It is a means to formally represent the interpretations of information found in normative text in frames. The original version of Flint distinguished three types of frames: institutional act frames, which builds upon the Power–Liability concept pair of Hohfeld, duty-claimright frames, and institutional fact frames. However, in recent work duty frames are recognised as a special sub-type of fact frames. As a result, the Flint language currently distinguishes only the act and fact frame (van Gessel et al. 2023). In this paper, we focus on the act frame.

Act frames are used to express a normative action performed by an actor upon an object to the benefit or detriment of a recipient. An example act frame is given in Table 1. 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. The *object* slot contains the thing that is acted upon, i.e. 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 precondition(s) and postcondition(s). A precondition must be satisfied to make the act valid. In the example given, the precondition to collecting personal data is that the personal data are processed lawfully, fairly and in a transparent manner in relation to the data subject. A postcondition represents how performing the act affects the normative state. Flint distinguishes creating and terminating postconditions, that create or terminate a previously non-existent or false fact respectively. In this example, a fact is created saying the controller shall be able to demonstrate compliance with Art. 5(1) of the GDPR.

Currently, there are four tools available to support a norm modeller in their endeavour (van Gessel et al. 2023).[2] First, the Source Decomposition tool, which is a tool that converts normative texts into RDF representations. Second, the Norm Editor, which can be used to create interpretations of norms using Flint, either manually or semi-automatically. Third, Automated Assistance, is a module on top of the editor that generates annotation suggestions based on

[2] https://gitlab.com/normativesystems/ui/interpretation-editor

the work of Bakker et al. (2022a) and van Drie et al. (2023). Finally, there is a software ecosystem that combines the tooling, and includes support to use a data repository of choice. The work described in this paper could improve and extend the Automated Assistance module.

2.2 Natural language processing

The field of Natural Language Processing (NLP) has seen considerable progress over the past decade, driven by advancements in neural networks and innovative techniques like word embeddings (Mikolovet et al. 2013). With the introduction of word2vec (Mikolovet et al. 2013), interest in distributional semantics methods increased. The parallel advancements of the broader field of machine learning led to substantial improvements in performance on standard tasks such as Question Answering and Named Entity Recognition. A turning point occurred in 2018 with the introduction of BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al. 2018), which utilises a transformer architecture (Vaswani et al. 2017) for language modelling. This architecture uses information from the surrounding sentence to generate contextual embeddings, capturing the meaning of a word or token based on both its preceding and following context. In 2020, Brown et al. (2020) introduced the large language model GPT-3 (Generative Pre-trained Transformer 3), which, like its predecessors GPT 1 and 2, uses an autoregressive architecture. Unlike BERT, GPT-3 employs an autoregressive architecture, considering only the left context when making predictions. This design choice, while limiting bidirectional understanding, allows for improved performance in language generation tasks and the ability to handle variable-length inputs more effectively. Brown et al. (2020) showed competitive results using few-shot learning on standard tasks such as Question Answering, showing the potential of these models. However, challenges have also been identified such as lack of semantic coherence, biases, and lack of transparency (Dale 2021). In this section, we elaborate on different applications and strengths of BERT and GPT models.

2.2.1 BERT

BERT is a transformer model, a deep learning architecture based on the multi-head attention mechanism, proposed by Vaswani et al. (2017). Specifically, BERT (Devlin et al. 2018) uses an encoder-only transformer architecture, that represents sequences of tokens—words or part of words—as numerical vectors. BERT is pre-trained on two tasks: masked token prediction and next sentence prediction. In masked token prediction, some tokens in the input are hidden, and the model learns to predict these hidden tokens based on their context. In next sentence prediction, the model is trained to determine the correct order of two spans of text by predicting which one appears first. This pre-trained model can be fine-tuned to perform a specific task, known as a *downstream task*, using annotated data. Fine-tuning is derived from transfer learning, an idea that was introduced for training neural networks (Bozinovski and Fulgosi 1976). The knowledge that is gained by a model

trained to perform one task can be leveraged to create a new 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 fine-tuned to recognise images of tables. Similarly, BERT models can be trained and fine-tuned in various ways according to the domain and the type of task. For instance, performance can be improved by pre-training on domain specific data (Zheng et al. 2022; Lee et al. 2019; Beltagy et al. 2019). These models are intended to address the fact that many terms and phrases have a different or unique meaning in a domain-specific context than they would have in a general context (Shaghaghian et al. 2020). For example, the word *attorney* occurs in both general and legal context, whereas *memorandum* is mostly used in the legal domain. Moreover, domain-specific texts, especially legal texts, often have a different syntactic structure than general domain texts. In the legal domain, the LEGAL-BERT collection of models is available (Chalkidis et al. 2020), which is pre-trained on a dataset of 12GB of English legal data from different subdomains such as contracts, legislation, and court cases.

Besides specific domain-models, other types of BERT models are trained on specific prediction tasks. For example, SpanBERT (Joshi et al. 2020) was introduced to improve performance on predicting spans of text. This model was not trained for a specific domain. In pre-training, instead of randomly masking 15% of the tokens in the pre-training text as BERT does, SpanBERT masks random spans of tokens in the text, thereby also masking 15% of the tokens in total. Thus, this 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. This means that it is more suitable for the prediction of longer spans.

Another specialisation of these models is a multilingual model. The authors of BERT also released Multilingual BERT (M-BERT) (Devlin et al. 2018), which is pre-trained on multiple languages. The training corpus for this model consisted of the combination of the Wikipedia content of the 104 most common languages. 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.

2.2.2 GPT

With the introduction of ChatGPT, the concept of Large Language Models (LLMs) became well-known in- and outside the academic world. Emergent behaviour of sufficiently large language models had been described by Wei et al. (2022), which became more apparent in all the potential ways users utilised ChatGPT that it had not been trained on specifically. Risks have been pointed out in literature and in popular media, but there are also opportunities such as increased efficiency due to its lack of needing annotated data, and its improved accuracy in several NLP tasks (Deng and Lin 2022). Besides ChatGPT, many other models exist, trained on different data (quality and amount), with pre-processing (such as filtering and tokenisation) and different architectures

(encoder-decoder, causal decoder, prefix-decoder and other) (Zhao et al. 2023). A few popular publicly available open source examples include BLOOMZ (Muennighoff et al. 2022, 2023), LLaMA 2 (Touvron et al. 2023), MPT (MosaicMLNLPTeam 2023), and Falcon (Almazrouei et al. 2023). Other popular models which are not publicly available include, but are not limited to: Gemini (Pichai and Hassabis 2023) and Claude 2 (Anthropic 2023).

Interaction with a GPT model primarily occurs through user interactions via ChatGPT, employing question-answer exchanges or dialogue sessions. Another method involves prompt engineering, where users, in this case more often researchers or data scientists, craft and furnish tailored texts to direct a model in generating desired responses or accomplishing specific tasks (Zhao et al. 2023). Prompt engineering can include specific instructions for the model as an assistant, and examples of outputs for the user. This technique which provides examples of input and output is known as few-shot learning (Brown et al. 2020; Gao et al. 2020), and has been shown to improve performance compared to zero-shot learning (just a prompt), reaching competitive results (Agrawal et al. 2022). Another option to steer the LLM model is function calling. Here, a function in the form of a dictionary prescribes the output format that the model should follow, thereby steering towards a more consistent output. This is important when output is reused as part of a pipeline, or for automatic evaluation (Eleti et al. 2023). However, guiding LLMs to follow an output schema is still a challenge, which results in a lack of quantitative evaluation on tasks such as semantic role labelling as Agrawal et al. (2022) point out.

Previous work has shown the potential of large language models for not only natural language generation tasks, but also for natural language understanding such as semantic role labelling. For example, Schucher et al. (2021) shows how prompt tuning can be used for semantic parsing. Using LLMs for such a task would limit the need for fine-tuning and thus for annotated data, thereby making natural language understanding tasks more accessible for smaller use cases.

2.3 Natural language processing for legal texts

In the legal domain, textual information is available in abundance. Within the field of Legal NLP, a diversity of tasks can be found, including machine summarisation, pre-processing, classification, information retrieval, information extraction, text generation, and creating or using resources such as taxonomies, datasets and code libraries (Katz et al. 2023). Our work falls in the information extraction category, and specifically in labelling.

Early work in the field of legal information extraction was done in the SALOMON project by Uyttendaele et al. (1998), which had the goal to automatically summarise legal texts to make them more accessible. Work on Italian law texts was done by Biagioli et al. (2005), who introduced a method for making the process of retrieving information from legal documents more efficient. They developed a system, SALEM (Semantic Annotation for LEgal Management), that automatically enriches law texts with semantic annotations on paragraph and

sentence level. Brighi et al. (2008) adopts a similar rule-based approach 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 to analyse the syntactic structure of modificatory provisions. This analysis 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 rule-based approaches to machine learning approaches. For example, the work by Gao and Singh (2014) implements an approach to automatically extract norms from contracts for multi-agent systems. With the development of the BERT (Devlin et al. 2018) and transfer learning, language models have been used increasingly for the task of modelling legal text. Shaghaghian et al. (2020) experimented with BERT-based models to perform document review tasks on legal texts. A recent study by Bakker et al. (2022a) uses Flint frames to represent information extracted from Dutch legal texts and considers a rule-based as well as a transformer-based approach to fill these frames. They compared a fine-tuned BERT model on annotated data to a rule-based approach, based on POS- and dependency tags. The fine-tuned model performed much better at 81%, which was later improved with an extended dataset to 84% by van Drie et al. (2023).

First work on generative LLMs for the legal domain explores tasks such as automatic annotations (Gray et al. 2023) and rule classification (Liga and Robaldo 2023). Savelka (2023) explore the LLMs' capacity for sentence-level annotations with zero-shot learning. They prompt the LLM with short text snippets, and the semantic types that must be used for the annotations, such as *evidence* or *legal rule*, together with compact definitions of these types. Although the model (GPT-3.5) was outperformed by fine-tuned smaller models, it still reached promising scores for categories such as *evidence* (micro-F1 of 0.77, compared to a micro-F1 of 0.91 for the fine-tuned model). While not suitable for fully automatic annotation due to their lower performance on other categories, workflows could be improved where resources for manual annotations are scarce (Savelka 2023).

3 Method

In this paper, we compare different approaches of recognising Flint roles in a sentence. We specifically focus on the roles in an act frame that are often present within a sentence, which are the roles actions, actor, objects and recipient. Preconditions and postconditions, on the other hand, are often found across the sentence boundary. This choice is consistent with the Dutch dataset we will be comparing this dataset with from van Drie et al. (2023). We introduce a dataset annotated with Flint roles (action, actor, object and recipient) used to fine-tune a set of language models: a base model, task specific models, domain specific models, and a multilingual model. The dataset is discussed in Sect. 3.1, the models in Sect. 3.2. Additionally, we query a non-fine-tuned generative LLM in four

different ways to assess the impact of task-specific annotated data on a complex task. This will be described in Sect. 3.2.4. Finally, we will explain the evaluation methods for all approaches in Sect. 3.3.

3.1 Data

We gathered sentences from five different EU regulations for the annotations and the subsequent fine-tuning of the models. These regulations contain many actions that individuals and organisations can and cannot take, and influence all individuals in Europe, making them an interesting use case for extracting act frame roles. The regulations are the General Data Protection Regulation (GDPR), the Digital Services Act, the Digital Markets Act, the Capital Requirements Act, and the Food Safety Act, which were collected from the EUR-Lex website.[3] To collect the sentences from the GDPR, we used the dataset with CLAL annotations from Nazarenko and Lévy (2021). This resulted in a total of 6664 sentences from EU regulations. After collecting all sentences, we manually selected the sentences together with a domain expert that contain an action. During this filtering, we looked for actions that would specifically influence or shape the behaviour of individuals or organisations, which is known as a normative effect (van Doesburg 2017; van Binsbergen et al. 2020). This concept was recommended as a selection criterion by van Drie et al. (2023). The filtering process resulted in 1575 action sentences.[4]

After collecting the sentences, five annotators labelled four Flint act-frame roles in each of the action sentences: action, actor, object, and recipient. The annotations were conducted based on a set of annotation instructions, which are included in appendix A. The annotation process resulted in 2335 annotated sentences. We computed the inter-annotator agreement for 196 unique sentences that were annotated by all annotators on the token level. We used Fleiss' kappa metric (Fleiss 1971) to determine the inter-annotator agreement. To compute the agreement per category, we implemented the specific agreement coefficient, which was first introduced by Dice (1945). The agreement scores can be found in Sect. 4.1.

For the sentences that were annotated multiple times by different annotators, we kept one unique annotation per action sentence to ensure data consistency. 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. Subsequently, we split the dataset for training, validation, and testing, using a 80-10-10% split for the training of all models. 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. We used the same 10% test split of sentences to test all our models.

[3] https://eur-lex.europa.eu/homepage.html

[4] Dutch data, BERT models and LLM approaches: https://gitlab.com/normativesystems/flintfillers/flintfiller-srl/-/releases/v3.1.0, English data and transformer models: https://gitlab.com/normativesystems/flintfillers/flintfiller-english

To compare the results between English and Dutch, we used an existing dataset and model[5] of 4463 annotated sentences from Dutch laws, created by van Drie et al. (2023). They fine-tuned their Dutch model on 90% of the 4463 annotated sentences. We reuse their dataset to fine-tune the Multilingual BERT model, and part of it as examples for the non-fine-tuned generative LLM. The inter-annotator agreement of the Dutch dataset is substantial ($\kappa = 0.75$). A ground truth was set aside of 104 sentences manually annotated by the authors, which we will use in this work as a test set.

3.2 Models

3.2.1 Rule-based 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 and the rules by Bakker et al. (2022b). Sentences were tagged with POS and dependency tags by the `en_core_web_sm` model from the spaCy toolkit,[6] chunks were then formed from the POS tags, and rules were applied based on these tags to assign the relevant Flint roles.

The rules are applied per sentence:

1. If a token has a *nsubj* or *obl:agent* tag, the phrase will be assigned the *actor* role.
2. If a token has a *dobj* or *nsubjpass* tag, the phrase will be assigned the *object* role.
3. If a token has a *dative* tag, the phrase will be assigned the *recipient* role.
4. If a token has a *root*, *ccomp* or *xcomp* tag, the 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*.

3.2.2 Mapping model

The standard SRL task is similar to the task of labelling Flint roles in a sentence. Therefore, we tested the performance of a model pre-trained on a SRL task by mapping the resulting semantic roles to Flint roles. First, the model labels the legal text that we provide as input. Next, the resulting semantic roles are mapped to thematic roles, which are then again mapped on the Flint roles. We used the pre-trained model from the `transformer-srl` library[7] to obtain the PropBank semantic roles (Palmer et al. 2005) for the sentences. Next, we used the mapping provided by VerbAtlas (Di Fabio et al. 2019), which contains a mapping from PropBank semantic roles to VerbAtlas semantic roles for 5306 PropBank predicate senses. We applied

[5] https://gitlab.com/normativesystems/flintfillers/flintfiller-srl/-/releases/v2.0.0

[6] https://spacy.io/..

[7] https://github.com/Riccorl/transformer-srl.

Table 2 Mapping from VerbAtlas thematic roles to Flint roles

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

this to all sentences whose 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.

The complete mapping from thematic roles to Flint roles that we utilised can be found in Table 2. Since this mapping follows an n-to-1 pattern, with no thematic roles mapping to more than one Flint role, it did not present any difficulties during our role translation process.

3.2.3 Fine-tuning models

As discussed in Sect. 2, fine-tuning pre-trained language models 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 pre-trained for a specific domain or task 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 our dataset on the Flint role labelling task:

- **BERT** Standard BERT model.
- **LEGAL-BERT** BERT model pre-trained on data from the legal domain. Potential to better understand patterns in our legal dataset.
- **EURLEX-LEGAL-BERT** BERT model pre-trained on data from EU regulations. Potential to better understand patterns in our legal dataset with EU regulations.
- **SpanBERT** BERT model pre-trained 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 described in the previous section. We fine-tuned Multilingual BERT, hereafter referred to as M-BERT, on the larger Dutch dataset (van Drie et al. 2023), to evaluate its ability to generalise to English for our labelling task. As described in

Sect. 2.2.1, this approach could possibly mitigate the need for separate datasets and model training for each language.

To monitor the performance of our models during fine-tuning, we used the cross-entropy loss function. The Dutch dataset by Bakker et al. (2022a) showed some imbalance across the Flint roles. We also observed such an imbalance in our dataset, with 52% of the tokens labelled as *O*, 28% as *Object*, 9% as *Actor*, 6% as *Action*, and 5% as *Recipient*. To avoid a bias toward the majority classes, we implemented class weighting by giving a set of weights to the loss function that ensured that the loss function prioritises 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. We fine-tuned each model for 4 epochs. Moreover, for the learning rate and batch size, we executed a grid search to systematically explore the best combination of values. For comparing the performance to Dutch, we reuse the fine-tuned BERTje model described by van Drie et al. (2023). BERTje (de Vries et al. 2019) is a Dutch version of BERT.

3.2.4 Generative LLMs

We use two generative LLMs to label the legal texts with the Flint roles. Due to its superior performance on benchmarks and additional functionalities such as function calling, we use the GPT-3.5 Turbo and the GPT-4 models[8,9]. We used a temperature of 0 to ensure a consistent output. To prompt the models, we use four different approaches: zero- and few-shot learning, function calling, and a combination of few-shot learning and function calling.

Zero-shot and Few-shot Learning

Without a fine-tuned model for a specific task, classification tasks can still be successful as Brown et al. (2020); Alex et al. (2021) have demonstrated. Generative LLMs such as GPT-3.5 Turbo have shown that with prompt engineering and few-shot learning, tasks such as text completion and code support are performed well. For our first approach to query the generative LLMs, we explore what results can be achieved on the task of classifying the different roles in a Flint frame with zero-shot and few-shot learning. We explored different prompt formulations, as well as differences in input format, such as tokenised versus non tokenised sentences. Preliminary results showed that prompts performed better on tokenised sentences.[10] Ultimately, we based our prompts on the annotation instructions of our English dataset (in appendix A), and of the dataset from van Drie et al. (2023). The prompt is shown in Table 3.

[8] https://platform.openai.com/docs/models/gpt-3-5.

[9] As described in the related work section, numerous models are available. While an exhaustive comparison is beyond the scope of this paper, detailed analyses can be found in various sources, primarily in blogs and preprints due to the rapidly evolving landscape.

[10] Due to the informal nature of these experiments and the absence of rigorous documentation, the resulting data is not included but can be requested from the authors.

Table 3 Example prompts and functions

Method	Prompt & function example
0-shot	Given a sentence: *sentence* Classify all elements into the following categories: Action, Actor, Object, Recipient or O. Only classify Actions, Actors, Objects and Recipients in the main clause. For tokens in subordinate clauses, use the category O. The amount of elements in the output list needs to equal the amount of elements in the input list. Generate all the elements of the output list. Do not omit anything
5-shot	*0-shot prompt* + *User*: "Verification of market prices and model inputs shall be performed by a person or unit independent from persons or units that benefit from the trading book.." *Assistant*: [Object, Object, Object, Object, Object, Object, Object, Action, Action, Action, Actor, Actor, Actor, Actor, Actor, Actor, Actor, Actor, Actor, Actor, Actor, Actor, Actor, Actor, Actor,..] *User*: "Power is delegated to the Commission to adopt the regulatory technical standards referred to in the first subparagraph in accordance with Articles 10." *Assistant*: [Object, Action, Action, Recipient, Recipient, Recipient, O,..]
Classification function	{name: classify_flint_labels, description: *0-shot prompt*, parameters: {type: object, properties: {action: {type: array, description: All words that are classified as the action of the sentence. An action consists of the main verb of the sentence and its auxiliaries and modals.}}}}
Masking function	{name: mask_flint_labels, description: Masks all action words in a sentence with [action]. Mask all actor words in a sentence with [actor]. parameters: {type: object, properties: {masked_sentence: {type: array, description: A sentence with all the action, actor, object, and recipient words of the sentence masked out with [action], [actor], [object], or [recipient].}}}}

Additionally to sending the instruction in the prompt, we add examples to explore how few-shot learning can improve zero-shot learning. Labelled examples are included in the request sent to the language model. Using this very limited set of annotated data, the model learns the task and the correct way of performing it. Results from Brown et al. (2020) show that this method is a promising alternative to fine-tuning a large language model, which is an expensive and time consuming task. For this experiment, five annotated sentences were randomly selected from the fine-tuned models' training set and incorporated into the prompt sent to the LLM. We provided the Flint roles as a list, as one would do in a classic token classification task. An alternative approach is to use a format where the sentence is interjected with the predicted roles such as Paolini et al. (2021) use. We decided on a list of roles because this aligns with our baseline models and fine-tuned BERT-based models. The LLM has not seen the examples given yet, and they are not contained in the test set, thereby ensuring a fair evaluation of the model's competencies. These examples were structured in a user-assistant format, as illustrated in the abbreviated example in Table 3. The full prompt containing all five examples is available in Appendix B.1.

Function calling

Output format matters when working with natural language understanding tasks. A downside of using generative LLMs for classification tasks is their inconsistent output (Agrawal et al. 2022). Flint roles are not the end product presented to users,

but it is rather a step which should support other parts of the Flint architecture (van Gessel et al. 2023), and therefore should be in a consistent format. Recently, several models introduced function calling functionality as a way to structure your request and the output of your request (Eleti et al. 2023). In this approach, we define a function for classifying the Flint roles in a sentence. We test two different functions: one for classification of the roles (`classify_flint_labels`), and a second one which masks the words in a sentence according to the right role instead (`mask_flint_labels`). Simplified versions of both functions are added in Table 3. In the classification function, the zero-shot prompt is used as description, whereas in the masking function an instruction is included to mask the words of a sentence by the label action, actor, object or recipient. The post-processed output of this approach will be evaluated with a mean accuracy score, as is done by Bakker et al. (2022a).

Classification function

The classification function takes a sentence and classifies the words or phrases from it that belong to the categories *actor*, *action*, *object*, and *recipient*. A shortened example containing only the action parameter is shown in Table 3. The full function can be found in appendix B.2. The output of this function is a dictionary with sets of words that are classified to these roles, and need to be post-processed to be evaluated. The classification function has a name, a description which includes a prompt from the zero-shot approach, and four parameters; one for every role type. Each of those parameters contain its type (an array) and a description. The classification function is passed to GPT, guiding the model to generate output in the specified JSON format, ensuring the correct mapping of words or phrases to their respective roles. For evaluating the results and comparing them to the other models, we use a post-processing script to go from the json output to a list of labels per word.

Masking function

We compare the classification function described above to a masking function, which masks the words in a sentence with the role it belongs to. A simplified example of the masking function is shown in Table 3. The full function can be found in appendix B.3. The function in this case only contains a description, and one parameter: the sentence with the words related to the roles masked. The description contains instructions for the model. The results of this function still need post-processing, but by directly marking roles in the text, hypothetically there is less risk of misclassification of double words.

Few-shot learning and function calling combination

For the last approach, we combined few-shot learning with function calling. Examples are given, and an output format is prescribed. This approach adds complexity to the prompt but has the benefits of both providing examples for the model to learn from, and prescribing an output format such that the results are reliable and can be evaluated.

3.3 Evaluation

To evaluate the performance of the fine-tuned models, we used precision, recall, the F1 scores per class, the macro F1 score, and the weighted F1 score. Moreover, considering the goal of filling Flint frames with complete roles extracted from legal text, 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 and Sundheim 1993). The MUC evaluation method allows us to distinguish whether models recognised a complete semantic role, or only part of it. Earlier work in Dutch from Bakker et al. (2022a), and van Drie et al. (2023) only included the accuracy score in their results. In the comparison between English and Dutch, we will also report the accuracy scores for the rule-based model and the mapping model.[11] To evaluate the output of GPT-3.5 Turbo and GPT-4, we post-processed the generated outputs to ensure they aligned with the required format, after which we calculated the accuracy and MUC-based metrics.

4 Results

This section presents the results from our methods described above. It is divided into four key subsections. In Sect. 4.1, the inter-annotator agreement from our annotated dataset is presented. In Sect. 4.2, we discuss the results from the rule-based model, the mapping model, and the fine-tuned English BERT model, and we compare them to the results for these methods in Dutch from previous work described by Bakker et al. (2022a) and van Drie et al. (2023). In Sect. 4.3, we will present the different evaluation metrics discussed above (Sect. 3.3) for all models that were fine-tuned on our dataset: the domain-specific models LEGAL-BERT and EURLEX-LEGAL-BERT (Chalkidis et al. 2020), the task-specific model, SpanBERT (Joshi et al. 2020), the multilingual model M-BERT (Devlin et al. 2018; Pires et al. 2019), and finally the standard BERT model (Devlin et al. 2018). Finally, in Sect. 4.4, we will show the results from querying GPT-3.5 Turbo and GPT-4 with the approaches described above.

4.1 Inter-annotator agreement

The annotations on the gathered English legal sentences resulted in an inter-annotator agreement of $\kappa = 0.712$, which is substantial agreement (Landis and Koch 1977), although a little lower than the inter-annotator agreement for the Dutch dataset (0.75) from van Drie et al. (2023). The results of the specific agreement for each category are shown in Table 4.

[11] Precision, recall, and F1 scores for these methods on our English dataset can be requested from the authors.

Table 4 The specific agreement (SA) per label for the annotated dataset

	Action	Actor	Object	Recipient
SA	0.938	0.946	0.713	0.708

Table 5 Accuracy for the rule-based model, mapping model, BERT for English and BERTje for Dutch

	Rule-based	Mapping	BERT(je)
EN	0.528	0.737	**0.879**
NL	0.587	–	**0.842**

The performance for the rule-based method and BERTje on Dutch is taken from van Drie et al. (2023)

4.2 Baseline models

We created a rule-based model and a fine-tuned BERT model for English and compare our results to earlier results for Dutch. We also created a mapping model that maps the results of a semantic role labelling model to Flint roles. This model exists only for English, as no semantic role labelling model is available for Dutch. Table 5 shows that the BERT model for English and BERTje model for Dutch have the best performance (indicated in bold). The mapping model, which was only available for English, does not outperform a BERT model which was fine-tuned on our specific task. Furthermore, the results show that the BERT model has a higher accuracy (0.879, also included in Table 6 in the next section) than the Dutch BERTje model (0.842), although BERT was fine-tuned on a smaller number of sentences

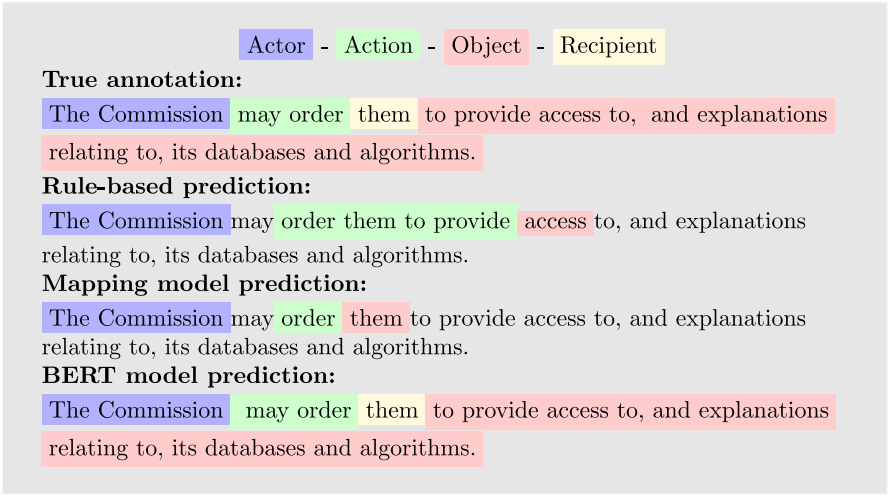

Fig. 1 An example sentence from the Digital Service Act, taken from the test set, with the true annotations and the predictions by the rule-based baseline and the mapping model

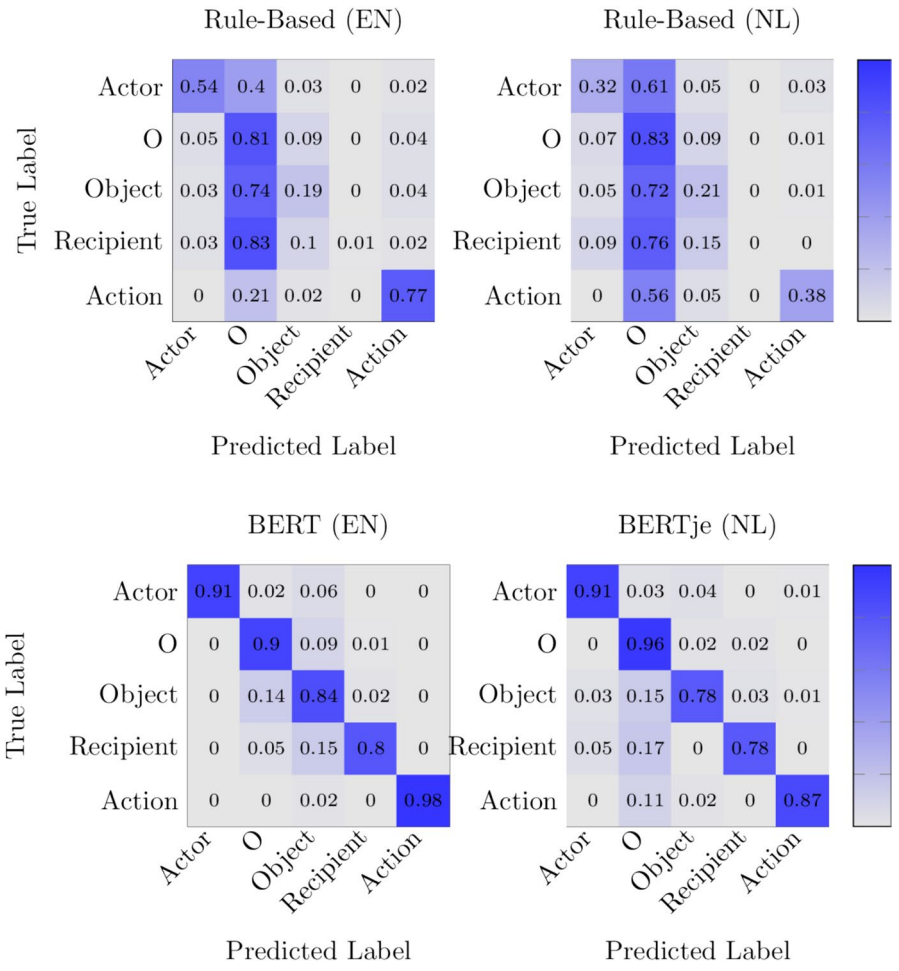

Fig. 2 Confusion matrices for baseline and BERT-based models

Table 6 Results of the evaluation on the token-level for fine-tuned BERT models

	BERT	M-BERT	SpanBERT	LEGAL- BERT	EURLEX- BERT
Accuracy	0.879	0.829	0.880	0.881	**0.896**
Balanced accuracy	0.885	0.890	0.890	**0.892**	0.885
Precision	0.888	0.830	0.880	0.911	**0.917**
Recall	0.885	0.832	0.890	0.892	**0.896**
Macro F1	0.886	0.829	0.884	**0.901**	0.900
Weighted F1	0.881	0.830	0.880	0.881	**0.896**

than BERTje (1868 and 4017 sentences for English and Dutch respectively). Finally, the rule-based model shows comparable performance for English and Dutch.

The confusion matrices in Fig. 2 show a similar pattern for the rule-based methods of both languages. First, they show that *O* (other) is predicted too often. The reason for this is that the rules in the rule-based method do not cover enough variation to classify tokens. Therefore, the default category *O* is chosen. Second, of the remaining categories, *actor* and *action* are classified with the highest performance. Finally, the matrices show that the rules did not cover the *recipient* category. In Fig. 1, an example sentence is given with the true annotation, and the predictions for the rule-based method, the mapping model, and the BERT model. In this example, the rule-based prediction does not contain the recipient, and predicts only one word of the object correctly, which explains its low performance on these roles. Figure 2 shows that the BERT-based models have a good performance for all categories. Moreover, they show that *Object* and *Recipient* are the categories with the lowest performance, as was the case for the rule-based method. In the example in Fig. 1, the BERT model predicts all words correct.

4.3 Fine-tuned BERT models

We fine-tuned five variations of BERT models on our dataset with Flint role labels, and evaluated them on token (word) level and on role level, as described in Sect. 3.3. The results on token level are presented in Table 6. The highest score per metric is indicated in bold. The results on all metrics are quite close together, but clear differences can be spotted between the Multilingual BERT model (M-BERT) and the other models. The Multilingual BERT model has a lower performance than the other models, but the results are still well above the rule-based and mapping approaches described in the previous section. SpanBERT's scores are very close to BERT's results, it performs a little higher on accuracy, balanced accuracy, and recall, whereas the standard BERT model had slightly higher scores on precision and both F1 scores.

Overall, we observe that LEGAL-BERT and EURLEX-LEGAL-BERT (shortened to EURLEX-BERT in Table 6), the models that were pre-trained on data from the legal domain before we fine-tuned them on our dataset, obtain the highest scores, although the differences are small in general. 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. Figure 3 reflects these results with an example. For this example sentence, the predicted labels for LEGAL-BERT en EURLEX-LEGAL-BERT are equal to the true annotation. The BERT prediction is almost correct; only 'the' is incorrectly labelled as an object. The example also shows the lower performance of Multilingual BERT (M-BERT) and SpanBERT, with the actor and the action being correct, but the recipient and object not.

In Fig. 4, the normalised confusion matrices for EURLEX-LEGAL-BERT and M-BERT models are shown, which contain the accuracy per class. It is notable that the performance is lower for the Object and Recipient classes, which are

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.

SpanBERT 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.

Fig. 3 An example sentence from the Digital Markets Act, taken from the test set, with the true annotations and the predictions by BERT, M-BERT, LEGAL-BERT, and EURLEX-LEGAL-BERT

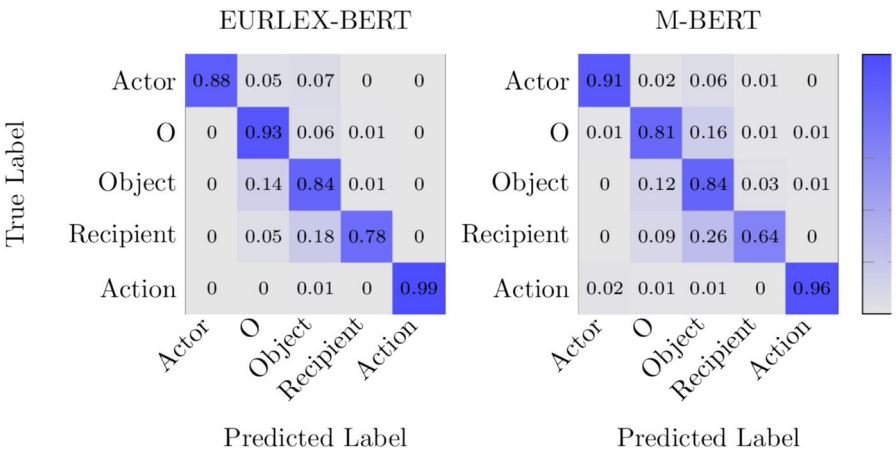

Fig. 4 Confusion matrices for EURLEX-BERT, and M-BERT

Table 7 Overall F1 scores on the MUC evaluation for the fine-tuned BERT models

	BERT	M-BERT	SpanBERT	LEGAL- BERT	EURLEX- BERT
Type	0.906	0.798	0.910	**0.924**	0.917
Partial	0.866	0.732	0.873	**0.888**	0.855
Exact	0.802	0.629	0.812	**0.833**	0.830
Strict	0.790	0.614	0.798	**0.822**	0.815

also the classes with the lowest inter-annotator agreement (see Table 4). Possible explanations include that the models struggled with ambiguity in the training labels, or that these roles are more complex in general and therefore harder to learn. We will discuss these results further in Sect. 5.

We also evaluated the fine-tuned BERT models on role level using the MUC evaluation (Chinchor and Sundheim 1993), as explained in Sect. 3.3. This evaluation takes partial correctness of the role into consideration. In Table 7, the overall F1 scores for the MUC evaluation are reported for all fine-tuned models on all four of the different evaluation schemas. The highest score per metric is again indicated in bold. These results exhibit a similar pattern as the results for the evaluation on the token level: the M-BERT model scores lower on all categories than the other models. The domain-specific models have the best performance, with LEGAL-BERT specifically scoring the highest on all schemas. Notably, in this case the EURLEX-LEGAL-BERT model scores lower than LEGAL-BERT. 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* 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 shows the highest F1-score at 82.2%.

4.4 GPT

We used four different approaches to prompt GPT-3.5-turbo and GPT-4: zero-shot and few-shot learning, function calling (with a classification and a masking function), and a combination of few-shot learning and function calling.

Table 8 Accuracy for GPT models on Dutch (NL) and English (EN) data

	GPT-3.5 Turbo					GPT-4				
	zero shot	Few shot	Cls	Mask	Combo	zero shot	Few shot	Cls	Mask	Combo
EN	0.080	0.026	0.558	0.530	**0.627**	0.175	0.272	0.498	0.557	0.586
NL	0.029	0.019	0.612	0.464	**0.644**	0.251	0.382	0.465	0.561	0.457

Table 9 Overall F1 scores on the MUC evaluation for the GPT Models for English

	GPT-3.5 Turbo					GPT-4				
	Zero	Few shot	Cls	Mask	Combo	Zero shot	Few shot	Cls	Mask	Combo
Type	0.401	0.498	**0.536**	0.203	0.375	0.511	0.498	0.330	0.505	0.332
Partial	0.320	**0.427**	0.353	0.183	0.288	0.392	0.385	0.293	0.335	0.298
Exact	0.150	**0.201**	0.077	0.028	0.059	0.182	**0.201**	0.183	0.074	0.106
Strict	0.145	0.193	0.069	0.017	0.054	0.176	**0.197**	0.173	0.068	0.095

Table 8 shows the accuracy for labelling the Flint roles in sentences for GPT-3.5 Turbo and GPT-4. The highest scores are indicated in bold. For both languages, the best performing generative LLM is GPT-3.5 Turbo with a combination of few-shot learning and function calling. This method has an accuracy of 0.644 for Dutch, and 0.627 for English. Of the different GPT-4 approaches implemented, the combination approach works best for English, and the function calling with masking function works best for Dutch. Overall, the worst performing approach is few-shot for GPT 3.5, and zero-shot for GPT-4 for both English and Dutch.

Figure 6 shows the confusion matrices for the combination method. The upper two confusion matrices show the best-performing generative LLM method, which is the combination method with GPT-3.5 Turbo. These figures highlight that the performance on the *action* category is relatively high, as is the performance on the *Other* category. The *recipient* category has the lowest performance. Another noteworthy source of errors is that too many tokens falsely received a prediction for the category *Other* (*O*). This general pattern can also be observed in the middle left confusion matrix, showing the combination method with GPT-4 for English. However, the confusion matrix for the GPT-4 combination approach for Dutch shows a rather different pattern. We observe that the relatively high accuracy obtained by this approach is mainly due to predictions in the *Other* category. Finally, the lower two confusion matrices in Fig. 6 show the performance of the zero-shot and few-shot approach with GPT-4 for English. While the accuracy indicates that the few-shot approach is better, the confusion matrix shows that the zero-shot approach is performing better in the *Other* category, while the few-shot approach is doing better for the other categories *Actor*, *Object* and *Recipient*.

The GPT-3.5 and GPT-4 models were also evaluated on role level using the MUC evaluation (Chinchor and Sundheim 1993), as detailed in Sect. 3.3. Table 9 presents the overall F1 scores across all four evaluation schemas, with the highest scores indicated in bold. Both models achieve their highest F1 scores on the *type* schema, where only correct role type assignment is required without precise boundary matching, while the *strict* schema—requiring exact boundary and type matches—yields the lowest F1 score.

To illustrate the difference in performance between the fine-tuned models and the GPT approaches, an example is given in Fig. 5. Whereas LEGAL-BERT predicts all labels correct (Fig. 3), GPT-4 (the combo method) confuses words in the dependent clause for an action and an act, and only correctly labels the

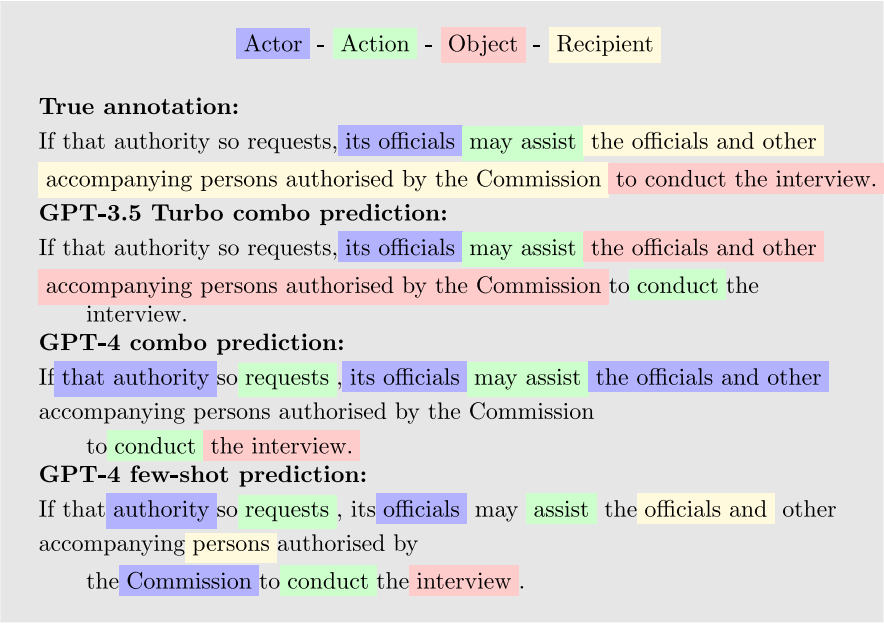

Fig. 5 An example sentence from the Digital Markets Act, taken from the test set, with the true annotations and the predictions by GPT-3.5 combo, GPT-4 combo and GPT-4 few-shot

main action and actor. GPT-3.5 Turbo correctly identifies the actor and action, but mislabels the recipient for the object. Our few-shot approach with GPT-4 performs worse. It only predicts parts of the roles correctly, but none of the roles are completely correct.

5 Discussion

The results of the rule-based model showed the lowest performance of all models except for the generative LLMs, followed consistently by the mapping model. This observation highlights the fact that the effectiveness of these approaches is directly tied to the quality and completeness of the components on which they are based. For the rule-based model this applies to the quality of the POS-tagger, the dependency parser, and mainly the rules that it relies upon, as Bakker et al. (2022b) also emphasise. As we saw in Table 5 in Sect. 4.2, the recall of the rule-based model 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.

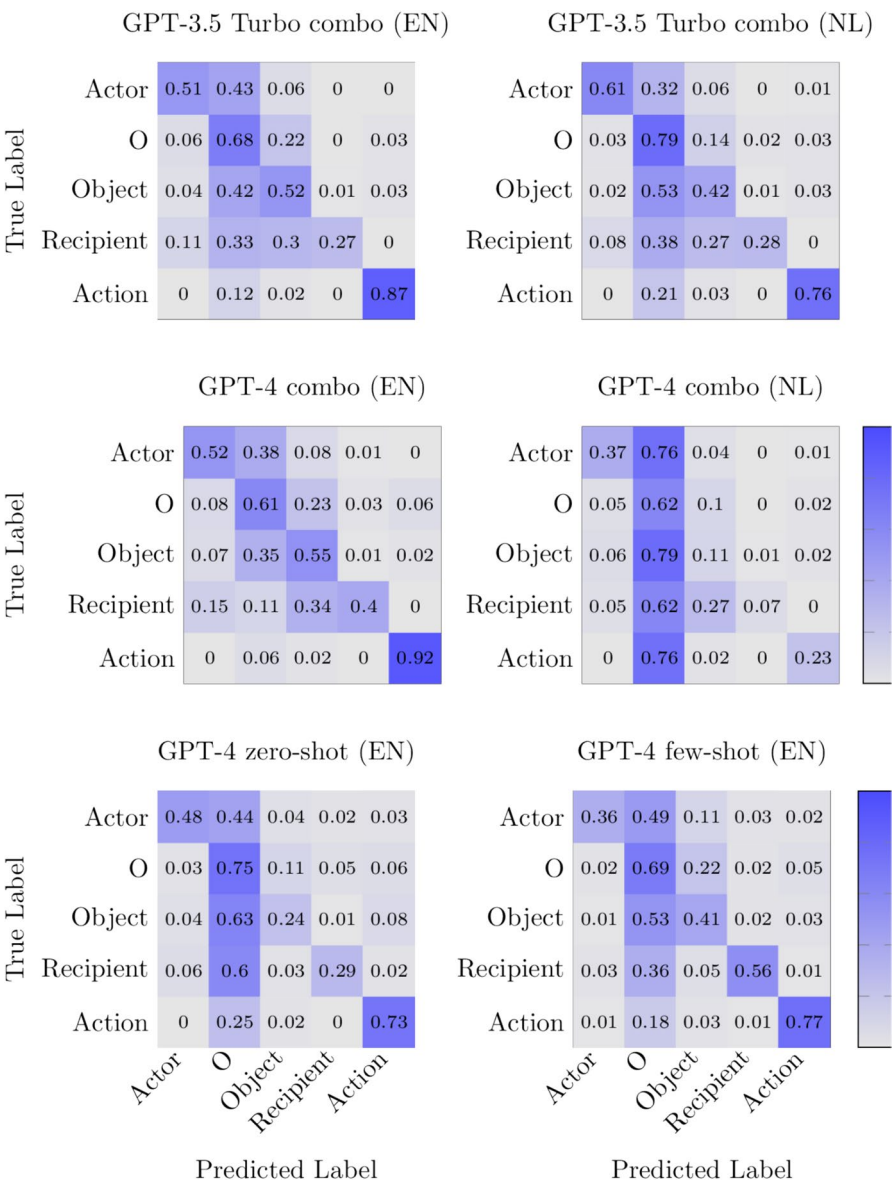

Fig. 6 Confusion matrices for the combination approach with GPT-3.5 Turbo and GPT-4 for English and Dutch, and the zero-shot and few-shot with GPT-4 for English

For the mapping model, the quality of the pre-trained SRL model that labelled text with PropBank roles, the mapping from PropBank to thematic roles, and the mapping from thematic roles to Flint roles all influence the quality of its predictions. The original SRL model on which the mappings were applied yielded an F1 of 86% on the CoNLL 2012 set (Pradhan et al. 2012), so

it could be expected that the performance on our data is similar at best, since the SRL model was trained on non-legal texts and legal texts are generally regarded as containing complex sentences. Moreover, the mapping from thematic roles to Flint roles is not necessarily completely correct due to the fact that both thematic roles and Flint roles are not universally defined. An alternative to the 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. However, at the time of this research, we are not aware of the existence of any pre-trained SRL models that use thematic labels in the legal domain.

When considering the results of the fine-tuned models, we saw that M-BERT's performance was relatively close to BERT's compared to the rule-based and mapping model, yet still performed significantly worse than BERT and all variations of BERT pre-trained on English data. 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 recognising the Flint roles. These roles are generally long and complex so the model may have had difficulties correctly identifying them in English, especially since it had not seen any specific examples of how these roles are usually structured in English. Its performance may be improved upon by combining datasets from different languages for fine-tuning, since M-BERT is also trained on multilingual data (Pires et al. 2019).

Overall, the domain-specific models achieved the best scores. The differences between LEGAL-BERT and EURLEX-LEGAL-BERT were small, despite EURLEX-LEGAL-BERT having been pre-trained on less legal data. However, we believe that the high performance of EURLEX-LEGAL-BERT can partly be attributed to the fact that our dataset solely consisted of sentences from EU legislation, which is exactly what EURLEX-LEGAL-BERT was pre-trained 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.

For the generative LLMs GPT-3.5 Turbo and 4, the results show that the different methods of prompting produce accuracy scores ranging from very low, such as with a zero-shot method, to higher, such as using a combination of few-shot learning and function calling. Interestingly, GPT-3.5 Turbo reaches higher accuracy scores than GPT-4 for the combination approach, although GPT-4 performs better at zero- and few-shot approaches. This is not in line with their performance on other tasks, where GPT-4 scores higher overall (OpenAI 2023). It might be interesting to explore whether these results are consistent over multiple runs, due to the inconsistency of these models and their tendency to hallucinate. The low scores of zero- and few-shot learning can partly be explained by the difference in answer formats. The models do not consistently output the same format and sometimes hallucinate extra tokens, or parts of answers such as an explanation. The scores for zero- and few-shot learning both are higher for the GPT-4 model. This might indicate that GPT-4 is better at providing

consistent answers in the same format, a trend supported by findings in other studies (Hackl et al. 2023). The few-shot results for GPT-4 are higher than the zero-shot, indicating that including examples in the prompt is beneficial for this model. We included five examples in our few-shot approach, though further research to determine the optimal number of examples or applying heuristics to improve their selection for this use case could enhance performance. Nevertheless, the function-calling approach achieved higher results, as it prescribes the output format, making automatic evaluation feasible. This is reflected in the accuracy scores: they went up for both approaches of function calling.

The differences in results from the two function calling approaches indicate that the choice of task type and prompting strategy influence model performance. The two types of functions show that GPT-3.5 Turbo has a slight preference for a classification task over a masking task; when asked to classify the words to certain roles, performance is higher than when asked to mask the words in the sentence. For GPT-4, it's the other way around. The combination approach with both few-shot learning and function calling yielded higher accuracy scores for English; few-shot learning improves the results by showing the model examples, while function calling leads to consistent outcomes without hallucinations. However, GPT-4 performed worse for Dutch on this method than on the function methods. This might be an indication that for GPT-4, different prompting strategies might be beneficial for this task. Another aspect in the prompt that can influence the results is the format that the model should output. While we adopted a token-based approach, alternative methods where phrases are kept together (e.g. (Paolini et al. 2021)) could be worth investigating further.

Regarding the performance of the generative LLMs on roles, higher scores are seen on the action role than the other roles. This implies that the action role is easier to recognise for the models, probably because the action is guaranteed to occur in every sentence due to the way our test sets were built. The lowest scores are found in the recipient role, followed by the object role. This is also the case for the fine-tuned models, although the differences are smaller. As Bakker et al. (2022a) describe, this might have to do with the length of the roles; the recipient and object can be long phrases or dependent clauses, whereas the action and the actor are often shorter. The *O* label is also more often predicted correctly, but that also has to do with the fact that anything not belonging to another role should be *O*, so there is a relatively high amount of *O* labels in the dataset. For the fine-tuned models, class weights were implemented to limit the influence of this imbalance. To further assess the influence of the imbalance and the generalisability of the models, applying cross-validation could be a valuable addition, computational resources permitting. For GPT-3.5 Turbo and GPT-4 in the few-shot and combo setting, the examples include more tokens with the *O* label, which might have influenced its performance on that label.

Interestingly, there are no big differences between Dutch and English. For the fine-tuned BERT models, the performance is a bit higher for English, even though the Dutch BERTje model is fine-tuned on more data. This can be explained by the overall better performance of the BERT model. No direct

comparison is made in the work of de Vries et al. (2019), but their NER accuracy scores are lower than the scores that BERT achieves (0.88 for BERTje versus 0.93 for BERT (Devlin et al. 2018)). For the rule-based model and GPT-3.5 Turbo and GPT-4, scores are a bit lower for the English data. This is surprising due to the fact that GPT models usually perform better on English than on other languages (OpenAI 2023).

Overall, fine-tuned models consistently outperform all different generative LLM approaches, both in general and across all roles. This can be attributed to the fine-tuned models having exposure to a larger number of examples and not relying on a prompt formulation with only a few examples. However, the advantage of generative LLMs like GPT-3.5 Turbo and GPT-4 lies in their ability to be queried without the need for an annotated dataset, yet still achieving reasonable performance depending on the prompt and approach used. Therefore, in resource-constrained settings, employing generative LLMs might still be advantageous as an initial step in extracting legal semantic roles. However, the results from the fine-tuned models indicate that specific tasks, such as this adaptation of semantic role labelling, still benefit significantly from the process of annotating data and fine-tuning a model when resources are available.

6 Conclusion

Norms are essential in our society, as they dictate how individuals should behave, interact, and conform within a community. Some norms are captured in laws, agreements, or other written sources. Interpretations of norms can differ, which is where formalisations such as Flint offer a solution. They express an interpretation of a norm in a transparent manner. However, creating these interpretations is labour intensive. Natural Language Processing (NLP) tasks such as semantic role labelling can support norm modellers. This paper explored whether different models can perform this task successfully. We implemented a rule-based model, based on the work of Bakker et al. (2022b), and a mapping model which maps thematic roles to Flint roles. We fine-tuned several BERT-based models among which two models pre-trained on domain specific texts. Lastly, we queried two generative LLMs with zero- and few-shot learning, function calling, and a combination thereof. To fine-tune the BERT-based models, we contributed a dataset of 1573 sentences from EU regulations with Flint act role annotations containing multiple annotations per sentence, with 2335 annotations in total. All models were evaluated and compared on their ability to correctly assign words in legal sentences to Flint act roles.

The results showed that SpanBERT, LEGAL-BERT, and EURLEX-LEGAL-BERT had the highest overall performance after they were fine-tuned on our dataset. On the token-level evaluation, EURLEX-LEGAL-BERT yielded the highest scores. On the role evaluation (MUC), SpanBERT, LEGAL-BERT, and EURLEX-LEGAL-BERT consistently outperformed BERT on all schemas, with LEGAL-BERT yielding the overall highest scores on all schemas. These results suggest that there is added value to using domain-specific and task-specific language models on this

particular task, with especially the domain-specific language models showing competitive results.

We fine-tuned M-BERT, a multilingual BERT model, on the Dutch dataset by van Drie et al. (2023), to explore the viability of re-purposing datasets from different languages for the same task in another language. After fine-tuning on the Dutch dataset and evaluating on the English test set, we found that whereas M-BERT outperforms the rule-based model and the mapping model, it does not manage to reach the level of BERT on our task. We conclude that more language-specific fine-tuning is required to learn these longer and more complex roles in legal text.

Additionally to fine-tuning these task-specific smaller language models, we tested several approaches for querying the GPT-3.5 Turbo and the GPT-4 model. If such generative LLMs are successful in this, it greatly reduces the need for annotated data, thereby reducing the reliance on labour-intensive manual annotation efforts and potentially accelerating the development and deployment of natural language processing applications. However, the results showed that the generative LLMs, GPT-3.5 Turbo and GPT-4, performed poorly on this task, especially when compared to the fine-tuned domain specific models such as LEGAL-BERT and EURLEX-LEGAL-BERT. This suggests that for complex natural language processing tasks, like our specialised form of semantic role labelling, the investment in domain-specific fine-tuned models through annotation efforts proves to be valuable.

For future research, it would be interesting to explore improving the performance of multilingual BERT (M-BERT) by combining datasets from different languages, for instance the Dutch dataset from van Drie et al. (2023) and the English dataset introduced in this work. Even though it did not perform as well as the other fine-tuned models, this method enables usage of resources from other languages when these are not available for the target language. Furthermore, the results of the generative LLMs GPT-3.5 Turbo and GPT-4 might be improved by narrowing down the task. For instance, one could simplify the task by asking for only one role at the time and combining them afterwards, potentially improving accuracy, but at the cost of it being more computationally intensive. Additionally, the influence of the functions that are used should be further explored; the difference between the two functions described above indicate that this can play an important part in the quality of the results. Another direction for future research is identifying the best k value for few-shot examples, as this could further optimise performance. Finally, large language models could also work together in a pipeline where humans combine and utilise output. An architecture where human knowledge and reasoning is combined with the capabilities of language models might be the solution to an efficient and transparent process of creating legal interpretations.

Appendix A Annotation instructions

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 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]$_{\text{ACTOR}}$.

Appendix B GPT function calls and few-shot learning examples

B.1 Few-shot learning examples

```
{"role": "system", "content": "You are an assistant that can tag the
↪  roles action, actor, object, and recipient in sentences. You tag
↪  words that do not fall into this category with o for other."}
{"role": "system", "name": "example_user", "content": "Significant
↪  subsidiaries of EU parent financial holding companies or EU parent
↪  mixed holding companies and those subsidiaries which are of material
↪  significance for their local market shall disclose the information
↪  specified in Articles 437 , 438 , 440 , 442 , 450 , 451 and 453 on
↪  an individual or sub- consolidated basis . "}
{"role": "system", "name": "example_assistant", "content": "['Actor',
↪  'Actor', 'Actor', 'Actor', 'Actor', 'Actor', 'Actor', 'Actor',
↪  'Actor', 'Actor', 'Actor', 'Actor', 'Actor', 'Actor', 'Actor',
↪  'Actor', 'Actor', 'Actor', 'Actor', 'Actor', 'Actor', 'Actor',
↪  'Actor', 'Actor', 'Actor', 'Actor', 'Action', 'Action', 'Object',
↪  'Object', 'Object', 'Object', 'Object', 'Object', 'Object',
↪  'Object', 'Object', 'Object', 'Object', 'Object', 'Object',
↪  'Object', 'Object', 'Object', 'Object', 'Object', 'O', 'O', 'O',
↪  'O', 'O', 'O', 'O', 'O', '.']"}
{"role": "system", "name": "example_user", "content": "Power is
↪  delegated to the Commission to adopt the regulatory technical
↪  standards referred to in the first subparagraph in accordance with
↪  Articles 10 to 14 of Regulation (EU) No 1093/2010 . "}
{"role": "system", "name": "example_assistant", "content": "['Object',
↪  'Action' 'Action', 'Recipient', 'Recipient', 'Recipient', 'O', 'O',
↪  'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O',
↪  'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O',
↪  '.']"}
{"role": "system", "name": "example_user", "content": "For the purposes
↪  of calculating own funds on an individual basis and a
↪  sub-consolidated basis , institutions subject to supervision on a
↪  consolidated basis in accordance with Chapter 2 of Title II of Part
↪  One shall not deduct holdings of own funds instruments issued by
↪  financial sector entities included in the scope of consolidated
↪  supervision , unless the competent authorities determine those
↪  deductions to be required for specific purposes , in particular
↪  structural separation of banking activities and resolution planning
↪  . "}
{"role": "system", "name": "example_assistant", "content": "['O', 'O',
↪  'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O',
↪  'O', 'O', ',', 'Actor', 'Actor', 'Actor', 'Actor', 'Actor', 'Actor',
↪  'Actor', 'Actor', 'Actor', 'Actor', 'Actor', 'Actor', 'Actor',
↪  'Actor', 'Actor', 'Actor', 'Actor', 'Actor', 'Actor', 'Action',
↪  'Action', 'Action', 'Object', 'Object', 'Object', 'Object',
↪  'Object', 'Object', 'Object', 'Object', 'Object', 'Object',
↪  'Object', 'Object', 'Object', 'Object', 'Object', 'Object',
↪  'Object', ',', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O',
↪  'O', 'O', 'O', ',', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O',
↪  'O', '.']"}
{"role": "system", "name": "example_user", "content": "Where an
↪  institution fails to meet the condition in point (b) of paragraph 1
↪  it shall immediately notify the competent authority . "}
{"role": "system", "name": "example_assistant", "content": "['O', 'O',
↪  'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O',
↪  'O', 'Actor', 'Action', 'O', 'Action', 'Recipient', 'Recipient',
↪  'Recipient', '.']"}
{"role": "system", "name": "example_user", "content": "Verification of
↪  market prices and model inputs shall be performed by a person or
↪  unit independent from persons or units that benefit from the trading
↪  book , at least monthly , or more frequently depending on the nature
↪  of the market or trading activity . "}
{"role": "system", "name": "example_assistant", "content": "['Object',
↪  'Object', 'Object', 'Object', 'Object', 'Object', 'Object',
↪  'Action', 'Action', 'Action', 'Actor', 'Actor', 'Actor', 'Actor',
↪  'Actor', 'Actor', 'Actor', 'Actor', 'Actor', 'Actor', 'Actor',
↪  'Actor', 'Actor', 'Actor', 'Actor', 'Actor', ',', 'O', 'O', 'O',
↪  ',', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O',
↪  'O', '.']"}
```

B.2 Function call for classification

```
{"name": "classify_flint_labels",
"description":
    "Classify all words in a sentence as part of the action, actor,
    ↪  object, recipient, or other."
    "Include determiners, adjectives, prepositions, complementisers,
    ↪  negations and phrasal verbs."
    "Exclude adverbs."
    "An action consists of the main verb of the sentence and its
    ↪  auxiliaries and modals.",
"parameters": {
  "type": "object",
  "properties": {
    "action": {
        "type": "array",
        "items": {
            "type": "string"
        },
        "description":
        "All words that are classified as the action of the sentence."
        "An action consists of the main verb of the sentence and its
                    auxiliaries and modals.",
    },
    "actor": {
        "type": "array",
        "items": {
            "type": "string"
        },
        "description":
        "All words that are classified as the actor of the sentence. "
        "An actor consists of the main agent who interacts with the
        ↪  action in the sentence."
        "The actor is the volitional causer of the event",
    },
    "object": {
        "type": "array",
        "items": {
            "type": "string"
        },
        "description":
        "All words that are classified as the object of the sentence. "
        "An object is the direct object of the action."
        "The object is the the entity which is moved by the action."
        "The object is the entity undergoing the effect of the action",
    },
    "recipient": {
        "type": "array",
        "items": {
            "type": "string"
        },
        "description":
        "All words that are classified as the recipient of the sentence.
        ↪  "
        "The recipient is the benefactor of the action.",
    },
    "other": {
        "type": "array",
        "items": {
            "type": "string"
        },
        "description":
        "All words that are not classified into the other categories",
    }
  }
  }
}
```

B.3 Function call for masking

```
{
"name": "mask_flint_labels",
"description":
"Masks all action words in a sentence with [action]."
"Mask all actor words in a sentence with [actor]. "
"Mask all object words in a sentence with [object]."
"Mask all recipient words in a sentence with [recipient]."
"An action consists of the main verb of the sentence and
↪ its auxiliaries and modals."
"An actor consists of the main agent who interacts with
↪ the action in the sentence."
"An object is the direct object of the action."
"The recipient is the benefactor of the action.",
"parameters": {
    "type": "object",
    "properties": {
        "masked_sentence": {
            "type": "array",
            "items": {
                "type": "string"
            },
            "description": "A sentence with all the
            ↪ action, actor, object, and recipient words
            ↪ of the sentence masked out with [action],
            ↪ [actor], [object], or [recipient]."
        }
    }
  }
}
```

Acknowledgements The authors would like to thank the Dutch Ministry of the Interior and Kingdom Relations for the financial support of this research. Furthermore, we extend our sincere gratitude to the norm engineering team for their feedback and work on the annotation tool. Finally, we express our gratitude to all annotators of the datasets.

Declarations

Conflict of interest The authors have no conflict of interest to declare that are relevant to the content of this article.

References

Agrawal M, Hegselmann S, Lang H, Kim Y, Sontag D (2022) Large language models are few-shot clinical information extractors. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 1998–2022

Alex N, Lifland E, Tunstall L, Thakur A, Maham P, Riedel C.J, Hine E, Ashurst C, Sedille P, Carlier A, et al (2021) A real-world few-shot text classification benchmark. arXiv preprint arXiv:2109.14076

Almazrouei E, Alobeidli H, Alshamsi A, Cappelli A, Cojocaru R, Debbah M, Goffinet E, Heslow D, Launay J, Malartic Q et al (2023) Falcon-40B: an open large language model with state-of-the-art performance. Technical report, Technology Innovation Institute

Anthropic: Claude 2. Anthropic (2023)

Bakker RM, van Drie RAN, de Boer MHT, van Doesburg R, van Engers TM (2022a) Semantic Role Labelling for Dutch Law Texts. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference, 448–457

Bakker RM, de Boer MHT, van Drie RAN, Vos D (2022b) Extracting structured knowledge from Dutch legal texts: A rule-based approach. In: EKAW (Companion)

Beltagy I, Lo K, Cohan A (2019) SciBERT: Pretrained language model for scientific text. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3606–3611

Biagioli C, Francesconi E, Passerini A, Montemagni S, Soria C (2005) Automatic semantics extraction in law documents. In: Proceedings of the 10th International Conference on Artificial Intelligence and Law. ICAIL '05, pp. 133–140. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/1165485.1165506

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

Brighi R, Lesmo L, Mazzei A, Palmirani M, Radicioni D (2008) Towards semantic interpretation of legal modifications through deep syntactic analysis. 189:202–206. 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, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D (2020) Language models are few-shot learners. In: Advances in Neural Information Processing Systems, pp. 1877–1901

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

Chinchor N, Sundheim B (1993) MUC-5 evaluation metrics. In: Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993

Dale R (2021) GPT-3: What's it good for? Nat Language Eng 27(1):113–118

de Vries W, van Cranenburgh A, Bisazza A, Caselli T, van Noord G, Nissim M (2019) BERTje: A Dutch BERT model. arXiv preprint arXiv:1912.09582

Deng J, Lin Y (2022) The benefits and challenges of ChatGPT: An overview. Front Comput Intell Syst 2(2):81–83

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

Dice LR (1945) Measures of the amount of ecologic association between species. Ecology 26(3):297–302

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

Eleti Atty, Harris Jeff, Kilpatrick Logan (2023) Function calling and other API updates. Accessed: 15 August 2023 . https://openai.com/blog/function-calling-and-other-api-updates

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

Gao T, Fisch A, Chen D (2020) Making pre-trained language models better few-shot learners. arXiv preprint arXiv:2012.15723

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

Gray M, Savelka J, Oliver W, Ashley K (2023) Can GPT alleviate the burden of annotation?, 157–166

Hackl V, Müller A.E, Granitzer M, Sailer M (2023) Is GPT-4 a reliable rater? evaluating consistency in GPT-4 text ratings. arXiv preprint arXiv:2308.02575

Hohfeld W (1917) Fundamental legal conceptions as applied in judicial reasoning. Yale Law J 26(8):710–770

Joshi M, Chen D, Liu Y, Weld DS, Zettlemoyer L, Levy O (2020) SpanBERT: Improving pre-training by representing and predicting spans. Trans Ass Comput Linguistics 8:64–77. https://doi.org/10.1162/tacl_a_00300

Katz D.M, Hartung D, Gerlach L, Jana A, Bommarito II M.J (2023) Natural language processing in the legal domain. arXiv preprint arXiv:2302.12039

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 CH, 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

Leone V (2021) Legal knowledge extraction in the data protection domain based on ontology design patterns

Liga D, Robaldo L (2023) Fine-tuning GPT-3 for legal rule classification. Comput Law Security Rev 51:105864

Mikolov,T. Sutskever I, Chen K, Corrado G.S, Dean J (2013) Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems **26**

MosaicMLNLPTeam: Introducing MPT-30B: Raising the bar for open-source foundation models. Accessed: 2023-06-22 (2023). www.mosaicml.com/blog/mpt-30b Accessed 2023-06-22

Muennighoff N, Wang T, Sutawika L, Roberts A, Biderman S, Scao T.L, Bari M.S, Shen S, Yong Z.-X, Schoelkopf H, et al (2022) Crosslingual generalization through multitask finetuning. arXiv preprint arXiv:2211.01786

Muennighoff N, Wang T, Sutawika L, Roberts A, Biderman S, Le Scao T, Bari M.S, Shen S, Yong Z.X, Schoelkopf H, Tang X, Radev D, Aji A.F, Almubarak K, Albanie S, Alyafeai Z, Webson A, Raff E, Raffel C (2023) Crosslingual generalization through multitask finetuning. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 15991–16111. Association for Computational Linguistics, Toronto, Canada. https://doi.org/10.18653/v1/2023.acl-long.891 . https://aclanthology.org/2023.acl-long.891

Nazarenko A, Lévy F (2021) Annotation files of the CLAL GDPR annotation, version 2. Available on https://lipn.univ-paris13.fr/~fl/CLAL/V2-2022-09/index.html

OpenAI: GPT-4 Technical Report (2023)

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

Paolini G, Athiwaratkun B, Krone J, Ma J, Achille A, Anubhai R, Santos C.N.d, Xiang B, Soatto S (2021) Structured prediction as translation between augmented natural languages. arXiv preprint arXiv:2101.05779

Pichai S, Hassabis D (2023) Introducing Gemini: our largest and most capable AI model. Google

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

Pradhan S, Moschitti A, Xue N, Uryupina O, Zhang Y (2012) CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In: Joint Conference on EMNLP and CoNLL - Shared Task, pp. 1–40. Association for Computational Linguistics, Jeju Island, Korea

Savelka J (2023) Unlocking practical applications in legal domain: Evaluation of GPT for zero-shot semantic annotation of legal texts. arXiv preprint arXiv:2305.04417

Schucher N, Reddy S, Vries H (2021) The power of prompt tuning for low-resource semantic parsing. arXiv preprint arXiv:2110.08525

Schwartz S, Yaeli A, Shlomov S (2023) Enhancing trust in LLM-based AI automation agents: new considerations and future challenges. arXiv preprint arXiv:2308.05391

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

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

Touvron H, Martin L, Stone K, Albert P, Almahairi A, Babaei Y, Bashlykov N, Batra S, Bhargava P, Bhosale S, et al (2023) Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288

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

van Binsbergen T, Liu L-C van Doesburg R, van Engers T (2020) eFLINT: a domain-specific language for executable norm specifications. In: Proceedings of the 19th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences, pp. 124–136

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

van Doesburg R, van Engers TM (2019) Explicit interpretation of the Dutch Aliens Act. In: Proceedings of the Workshop on Artificial Intelligence and the Administrative State Co-located with 17th International Conference on AI and Law, pp 27–37

van Drie RAN, de Boer MHT, Bakker RM, Tolios I, Vos D (2023) The Dutch law as a semantic role labeling dataset. In: Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law, 316–322

Van Gessel T, Biagioni G, Breteler J, Tolios I, Boertjes E (2023) A toolset for normative interpretations in FLINT. In: SEMANTiCS (Posters & Demos)

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A, Kaiser L, Polosukhin I (2017) Attention is all you need. In: NIPS

Wei J, Tay Y, Bommasani R, Raffel C, Zoph B, Borgeaud S, Yogatama D, Bosma M, Zhou D, Metzler D, et al (2022) Emergent abilities of large language models. arXiv preprint arXiv:2206.07682

Zhao W.X, Zhou K, Li J, Tang T, Wang X, Hou Y, Min Y, Zhang B, Zhang J, Dong Z, et al (2023) A survey of large language models. arXiv preprint arXiv:2303.18223

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