Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 448–457
Marseille, 20-25 June 2022

Semantic Role Labelling for Dutch Law Texts

R.M. Bakker[1], R.A.N. van Drie[1], M.H.T. de Boer[1], R. van Doesburg[1,2], T. van Engers[1,2]
[1] TNO [2] Leibniz Center for Law, UvA
[1] Anna van Buerenplein 1, 2595 DA, The Hague, The Netherlands
[2] Postbus 1030, 1000 BA, Amsterdam, The Netherlands
{roos.bakker, romy.vandrie, maaike.deboer, robert.vandoesburg, tom.vanengers}@tno.nl

Abstract

Legal texts are often difficult to interpret, and people who interpret them need to make choices about the interpretation. To improve transparency, the interpretation of a legal text can be made explicit by formalising it. However, creating formalised representations of legal texts manually is quite labour-intensive. In this paper, we describe a method to extract structured representations in the Flint language (van Doesburg and van Engers, 2019) from natural language. Automated extraction of knowledge representation not only makes the interpretation and modelling efforts more efficient, it also contributes to reducing inter-coder dependencies. The Flint language offers a formal model that enables the interpretation of legal text by describing the norms in these texts as acts, facts and duties. To extract the components of a Flint representation, we use a rule-based method and a transformer-based method. In the transformer-based method we fine-tune the last layer with annotated legal texts. The results show that the transformed-based method (80% accuracy) outperforms the rule-based method (42% accuracy) on the Dutch Aliens Act. This indicates that the transformer-based method is a promising approach of automatically extracting Flint frames.

Keywords: semantic role labelling, part-of-speech tagging, Dutch, transformers, legislation

1. Introduction

Imagine you are an interpreter of the law with the task to determine if you could grant someone's application for a residence permit. To support your decision, you might look for similar cases, and you may find out that another application reviewer has made a different decision on a similar case than the one you had in mind. This difference may be for different reasons, including a different interpretation of the normative texts that were used to come to that decision. In this case, it would be helpful to have an explicit formal representation of the interpretation of the relevant norms in order to compare them and resolve interpretation conflicts. Flint (van Doesburg and van Engers, 2019) is a knowledge representation (KR) language that allows us to unambiguously express the meaning (interpretation) of sources of norms. Possible sources for formalisation are texts concerning the regulation of human behaviour, including legislation, policy guidelines, contracts and doctrine.

Expressing interpretations of sources of norms using KR typically is a labour-intensive, manual task. However, doing this automatically using Natural Language Processing (NLP) is challenging, as shown in previous studies (van Gog and van Engers, 2001; de Maat and van Engers, 2003; Winkels et al., 2005; de Maat et al., 2009). In these early attempts, the target KR only allowed for a limited form of reasoning. The approach developed at the time sped up the manual labour involved in creating computer-understandable knowledge representations. However, the attempts targeted one specific task, so the KR efforts were not optimally used. Moreover, a more powerful KR has since been developed: Flint (van Doesburg and van Engers, 2019). Flint is aimed to be a more generic and less task-dependent modeling language that can be used to express the interpretation of any source of norms. When modeling in Flint, we include the perspectives of all agent roles affected by the norms and focus on the action part of norms. As a result, the KR can be used in multiple task contexts using multiple forms of reasoning. We may, for instance, use the KR as a knowledge base for multidisciplinary teams, as specification for decision-support systems, to detect anomalies such as conflicts in duties or missing powers to fulfill duties, or to build eServices assisting citizens.

The work presented here builds upon previous work that studies how and to what extent NLP can be used to lighten the task of interpretation modeling. Specifically, we study how we can automatically extract Flint frames, which are the building blocks of the Flint language.[1] In this paper, we compare two NLP approaches to extract an interpretation from a Dutch legal text and express it in Flint. The first approach is similar to the approach by de Maat et al. (2009). We use part-of-speech tagging and chunk tagging, and then allocate chunks to the slots in a Flint frame using a set of rules. A rule-based approach is used in many of the earlier studies on automatic extraction of interpretation of norms. In recent years, transformers have become more prevalent in NLP, improving performance

[1] As there are often multiple possible interpretations of a source text, the goal of the NLP system is not to find the one true interpretation of a source. Rather, the goal is to suggest one possible interpretation which might serve as a starting point from which other interpretations deviate.

on many tasks. Therefore, our second approach is a transformer-based method. We fine-tune a transformer model for a token classification task similar to semantic role labelling, since the slots to be filled in a Flint frame are similar to semantic roles. We fine-tune the model with annotated legal texts, since to our best knowledge there currently does not exist a Dutch language model for semantic role labeling. We conducted an experiment to qualitatively evaluate and quantitatively compare the results of both approaches, i.e. these automatically extracted Flint frames.

2. Related Work

In this section we first discuss earlier work that used NLP to improve modeling of legal texts. The framework we use in our method, Flint frames, will be discussed in section 2.2. Our goal is to automatically extract Flint frames, and we do this using the technique semantic role labelling, which will be discussed in section 2.3.

2.1. NLP for normative texts

Relevant work has been done on Italian legislation, specifically on provisions. Francesconi (2006) divided these provisions into two subclasses: amendments, such as insertion and substitution, and rules, such as definitions and obligations. In the Flint language definitions are typically represented as fact frames, while obligations typically are expressed as duty frames.

Brighi et al. (2008) focused on amendments (also called modificatory provisions, or normative modifications). Such amendments are instructions to the reader expressing how a legal text has to be modified. They used a frame for each type of modification, such as deletion and substitution. To fill the frame, first they used a rule-based parser named TUP to extract the syntactic structure of sentences in dependency format. Then, they selected sentences with specific verbs indicating a specific provision, and filled the semantic slots of a frame using a rule-based approach.

Spinosa et al. (2009) also focused on provisions that specify a change to a legal text. Their goal was to automatically extract metadata describing the changes, including but not limited to the text to be changed and the text that is to be inserted. The preprocessing, which includes the detection of normative references in the text that have to be modified, was done with xmLeges tools (Agnoloni et al., 2007). They used AnIta NLP tools (Tamburini, 2012) for their linguistic analysis of the norms and to create the metadata describing the changes to the norm.

Biagioli et al. (2005) worked on provisions, both amendments as well as rules. Examples of rule provisions are obligation, permission and prohibition. The authors have created a provision automatic classifier (multi-class SVM) that detects the type of provision. They also proposed a provision argument extractor which outputs the semantic tags of the text. For example, an obligation has the arguments addressee, action and third-party. To create these semantic tags, they first performed a syntactic analysis which outputs part-of-speech tags, chunk tags and dependencies. Using this syntactic analysis, they did a rule-based semantic analysis to allocate the textual elements to the arguments of the provision.

The work mentioned above is focused on Italian legislation; NLP researchers have also worked on other languages, such as English and Dutch. Recent research on English legal text has mainly focused on the transition from semantic analysis to a more context-aware analysis, often using transformer models such as BERT. Elwany et al. (2019) and Chalkidis et al. (2020) for example, experimented with BERT pre-trained and fine-tuned on legal documents. They showed that a BERT model trained on domain-specific legal text can outperform a general pre-trained BERT model. On the other hand, Shaghaghian et al. (2020) explored different usages of BERT in four main scenarios in legal document reviewing. In their experiments they showed that on token level tasks, such as semantic role labelling, general domain pre-trained BERT models work better. However, on sentence level tasks, such as text similarity or rule navigation, customization of language models is beneficial.

NLP research within the Dutch legal text domain has mainly focused on syntactic and semantic analysis. van Gog and van Engers (2001) proposed a tool OPAL (Object-oriented Parsing and Analysis of Legislation) to support the modeling of legislation. The goal of OPAL is to model noun phrases and specific patterns in legal sentences in Unified Modeling Language (UML) and Object Constraint Language (OCL). OPAL contains a grammar, parser, lexicon and category guesser, all of which are rule-based. Using the inputs from these components, a model is generated.

de Maat et al. (2009) aimed to describe formal interpretations of Dutch law texts using OWL as knowledge representation language. Specifically, the goal was to extract norms, which have specific subtypes: obligations, rights, application provisions, penalisations, calculations, delegations and publication provisions. The preprocessing consists of parsing the text to extract sentences from Dutch law texts, which was done using the General Architecture for Text Engineering (GATE) and the JAVA Annotations Pattern Engine (JAPE). To create the norms, first, each sentence was classified to one of the above norm types. To do this, a pattern matcher and an SVM classifier were proposed. Second, the sentences were parsed using Alpino, a dependency parser for Dutch (Bouma et al., 2001). Third, the frames were extracted using rules, such as the subject is the agent of the action, the direct object is the patient of the action, and the indirect object is the recipient of the action. Finally, the frames were translated to an OWL format. While OWL has certain advantages over UML/OCL and, like any description logic, allows for some automated reasoning, i.e. automatic classification, OWL also has some serious limitations (Hoekstra, 2009). These reasons amongst others motivated us to look for a more expressive, task-independent KR formalism that is easy to understand by both human as well as computers, a quest that lead to the development of a new KR formalism: Flint.

2.2. Flint Frames

Flint (van Doesburg and van Engers, 2019) is a framework to represent sources of norms. It is a formal language consisting of three types of frames that are used to unambiguously express the interpretation of sources of norms: acts, facts and duties.

An act frame describes the normative actions that can be performed by an actor, if a certain precondition holds, and when performed has an effect on the agent bound by the action: the recipient. Act frames contain an action, actor, object, recipient, precondition and postcondition. The postcondition contains the result of an action, which can be the creation or termination of one or more facts or duties. An example of a simplified act is given in Table 1. It is simplified in the sense that only one precondition is given instead of the complete set of preconditions. The complete act frame can be found in the Appendix.

A duty frame describe acts that ought to be performed in the future or, in case of a violation, should have been performed in the past. A duty frame contains a duty holder and a claimant. It also has one or more creating acts that can create the duty, one or more enforcing acts that can be used to enforce the satisfaction of the duty in case the duty holder neglects his duty, and one or more terminating (or satisfying) acts that effectively terminate the duty.

A fact frame can be used to make detailed statements on the precondition of an act. It contains a function, which is either a Boolean that expresses the condition that makes a fact true, or an arithmetic function.

In this paper, we focus on the creation of Flint act frames. Several act frame components can be seen as semantic roles: the actor, object and recipient are all semantic roles of the action. Note that the reverse is not true: if we obtain actors, objects and recipients from a text, they are not necessarily part of an act frame. As stated before, act frames describe not any action but specific normative actions that have an effect in the real world, such as granting an application. However, we can obtain semantic roles of an action using semantic role labelling, which is the focus of one of our approaches, and the subject of the next section.

2.3. Semantic Role Labelling

Semantic role labelling (SRL) refers to the process of labelling chunks in a sentence with their semantic role, indicating who did what to whom. In recent years, neural network models have shown to be effective for the task of semantic role labelling. Several accounts show the added value of enriching the model with syntactic features (He et al., 2018a; He et al., 2018b; Strubell et al., 2018). Recently, however, a BERT-based model without additional features has been used to achieve state-of-the-art performance on semantic role labelling by fine-tuning BERT on an SRL task (Shi and Lin, 2019). This is in line with recent successes of BERT-based approaches in NLP. The original BERT model (Devlin et al., 2019) is trained on English, but a Dutch model called BERTje is also available (de Vries et al., 2019).

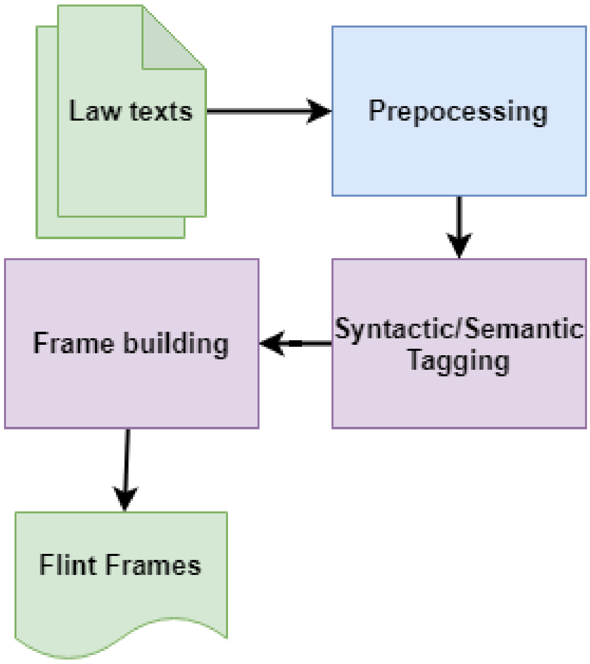

Figure 1: Architecture of the method

3. Method

We compared two NLP approaches - one based on the Dutch literature discussed in 2 and one based on the latest state of the art in SRL - to extract an interpretation from a legal text and express it in Flint. Figure 1 shows the architecture behind both approaches. First, law texts are gathered and preprocessed. This is discussed in section 3.1. Next, the sentences are tagged, either with syntactic or semantic tags depending on which approach is chosen. The first approach is rule-based, and results in syntactic tags. The rule-based approach is discussed in section 3.2. The second approach is based on a a machine learning method: since a Dutch language model on semantic role labelling does not exist, we fine-tuned a transformer model for a token classification task similar to semantic role labelling. This approach outputs semantic tags, and is discussed in 3.3. The data and the code are publicly available[2].

3.1. Data

The law texts are collected from the governmental website where all Dutch legislation is published, `wetten`.

[2] https://gitlab.com/calculemus-flint/flintfillers

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

Table 1: Simplified example of a manually created act frame from the GDPR

nl. For both of our approaches, some structural elements regarding the format of the text are removed, and the text is split up in sentences. No further preprocessing is needed for the rule-based approach, but the transformer-based method does need more preprocessing. Specifically, annotations are needed to fine-tune the transformer model for the semantic role labeling task.

To create an annotated dataset to fine-tune the model on a semantic role labelling task, five Dutch law texts are selected: the General Administrative Law Act ('Algemene wet bestuursrecht'), the Constitution ('Grondwet'), the Procurement Act ('Aanbestedingswet'), Accountability Act ('Comptabiliteitswet') and the Aliens Act ('Vreemdelingenwet'). The annotation task consisted of selecting those sentences that contain an act and then annotating the components of the act: the action, actor, object and recipient. The annotators received a detailed instruction on how to annotate the components of the act to guarantee uniformity in the annotated data. Explanations about the roles were also included in the instructions. The details of the instructions are shown in Table 2.

In total 1854 unique sentences were selected for annotation, of which 497 sentences contained an action. Only the sentences containing an action were used in the training of our transformer-based approach (positive sentences). The annotations on the first 315 sentences of the Aliens Act, of which 88 contain an action (of which 82 contain an object, 21 a recipient and 50 an actor), were kept separate as a control dataset while the other data is used for the fine-tuning process. Of the remaining 409 positive sentences, 324 contained an actor, 337 an object, and 85 a recipient. Fleiss' kappa was run on all annotations of sentences containing an action to determine if there was agreement between the annotators' judgements on the token classification task. The first three authors all contributed as annotators. Additionally, three interns were asked to annotate sentences on a voluntary basis, though they received an indirect reward in the form of an internship compensation. Annotators were given sentences at random, while ensuring that all sentences were given to two different annotators. Each annotator could select from one of five categories for each token of the sentence to be annotated: "action", "actor", "object", "recipient" and "other" (this category was reserved for all tokens that did not receive one of the first four categorizations). Fleiss' kappa showed there was substantial agreement between the annotators' judgements, $\kappa = .785$.

3.2. Rule-Based Method

For the rule-based method, we focus on syntactic analysis of the text. A downside of using Dutch text, and especially law text, is that only a few NLP tools have been designed for the purpose of processing this language because of the relatively small language community. Moreover, the performance of those tools is lower compared to those designed for processing English texts. We use a part-of-speech tagger, which outputs the grammatical role per word, and a chunk tagger using a light-weight dependency parsing method. With the combination of these two tools, we have an overview of the grammatical roles per sentence as well as the relations between words. For both tools, the implementation of Pattern[3] is used, because this tool can recognise law-related verbs such as *inwilligen* (grant). After the sentences are syntactically tagged, the following rules are applied to create Flint frames from these tags. Note that recipients / interested-party are not included in these rules since they are often not explicitly present, but may be inferred from other sentences [4].

- An action is taken from a list created by a domain expert of action verbs. If the infinitive version of the verb is present in the list (extracted using Pattern), the word is labelled as an action.
- an object is the first NP in the sentence in case of a passive sentence
- an actor is a word related to a Proper Noun (PRP or NNPS(S))

[3] https://pypi.org/project/Pattern

[4] In situations where this is not possible, the source contains implicit knowledge, because a legal act that binds no one is void. By including this implicit knowledge in an act frame this knowledge is made explicit.

Annotation Instruction	Example (Dutch)	Example translation
copy literal parts of the sentences		
include determiners	*het* bestuursorgaan	*the* administrative authority
include adverbs that are considered essential (often in an action)	*elektronisch* verzenden	send *electronic*
combine clusters of verbs (including auxiliary verbs)	Het bestuursorgaan *kan* van een gemachtigde een schriftelijke machtiging *verlangen*	The administrative authority *can require* a written authorisation from an authorised representative
include negations in the action	Het bestuursorgaan *gebruikt* de bevoegdheid tot het nemen van een besluit *niet* voor een ander doel dan waarvoor die bevoegdheid is verleend	The administrative authority *does not use* the authority to make a decision for another purpose other than which that authority has been granted
include all parts of separated verbs	Het bestuursorgaan *zendt* geschriften die niet voor hem bestemd zijn en die ook niet worden doorgezonden, zo spoedig mogelijk *terug* aan de afzender	The administrative authority returns (*sends back*) writings that are not intended for it and they (writings) are not forwarded, as soon as possible to the sender
do not include modifiers	Het bestuursorgaan zendt geschriften tot behandeling *waarvan kennelijk een ander bestuursorgaan bevoegd is*, onverwijld door naar dat orgaan, *onder gelijktijdige mededeling daarvan aan de afzender*	The administrative authority sends writings under consideration *of which another government body apparently is responsible for*, without delay to that government body, *while at the same time a notification of this to the sender*
do not include preconditions	Bestuursorganen en onder hun verantwoordelijkheid werkzame personen gebruiken de Nederlandse taal, tenzij bij wettelijk voorschrift anders is bepaald.	Government bodies and under their responsibility working people use the Dutch language, unless by law otherwise specified
actions within preconditions are not considered actions	Een bestuursorgaan kan een elektronisch verzonden bericht weigeren voor zover de betrouwbaarheid of vertrouwelijkheid van dit bericht onvoldoende *is gewaarborgd*, gelet op de aard en de inhoud van het bericht en het doel waarvoor het wordt gebruikt.	The administrative authority can refuse an electronically send message when (/in the extent) the reliability or confidentiality of this message *is* insufficiently *guaranteed*, regarding the nature and content of the message and the goal for which it is used.

Table 2: Annotation Instructions and Examples

3.3. Transformer-Based Method

For the transformer-based method, we use BERTje, a Dutch transformer model by de Vries et al. (2019). We fine-tune the last layer with our annotated dataset described in 3.1. The dataset includes all annotated data except the annotations on the first 315 sentences of the Aliens Act (our control dataset). The dataset is split randomly in 365 train sentences, 20 validation sentences and 21 test sentences. We did an ablation study on the sets of parameters based on the ones recommended in (Devlin et al., 2019) and (McCormick and Ryan, 2019). The highest accuracy (0.84)[5] was achieved with a learning rate of 5e-5 over 4 epochs, with a batch size of 8 and a weight decay of 0.01. The resulting model labels the test sets with *action*, *actor*, *object*, and *recipient*, which can then directly be used in their matching slots to build a Flint frame.

3.4. Evaluation

The performance of the transformer-based method is calculated in the training process described above. The performance of the rule-based method is calculated based on the full annotated dataset, using the mean accuracy score (number of correctly classified words in a sentence divided by all words, averaged over all sentences). A confusion matrix of the roles is also used to get insight in the mistakes made. To evaluate and compare both methods, the same metrics are used on two datasets: the test set of the annotated dataset and control dataset (the annotations on the Aliens Act). The next section will elaborate on the results of both meth-

[5] validation loss: 0.45, training loss 0.47, precision 0.39, recall 0.44, F1 0.41

Dataset	Rule	Transformer
Annotated dataset	0.42	-
Test set	0.52	0.81
Aliens Act (positives)	0.42	0.80
Aliens Act (all)	0.74	0.69

Table 3: Mean accuracy score for the different datasets and methods

ods.

4. Results

Table 3 shows the mean accuracy score for the different datasets for both the rule-based and the transformer-based method. The first dataset that is shown in table 3, the annotated dataset, contains all annotated data together. The accuracy for this dataset could only be calculated for the rule-based method since part of this data was also used for training in the transformer-based method. The second dataset, the test set, is one-third of the complete annotated dataset with which the model is fine-tuned (the other two sets being the training and the validation sets). The Aliens Act set was kept apart as a control dataset. The Aliens Act (positives) set contains only the sentences with an action in it, whereas the Aliens Act (all) dataset contains also sentences without an action (negatives). The results show that on both the test set and the Aliens Act the transformer-based method is almost twice as good as the rule-based method. If we do not only calculate performance on all positively annotated sentences, but all 315 annotated sentences from the Aliens Act (including sentences not containing acts), the performance of the rule-based method is higher compared to the transformer-based method.

Figure 2 shows the confusion matrix of the rule-based method on the test set. These results show that many semantic roles are missed, the predicted 'O' category has very high values for all true labels. None of the recipients are recognised, all values are 0.0 in that column, and the verbs that are recognised are classified relatively well (0.12 on the V-V block).

Figure 3 shows the confusion matrix of the transformer-based method on the test set. In the diagonal the correct annotations are shown. Verbs are classified correctly most of time (V-V block, 0.94), the recipient ('REC') is most often misclassified as other ('O'). The other roles have a correct classification between 0.70 and 0.84.

Figure 4 and 5 show the confusion matrices for the second dataset, the Aliens Act. These matrices show similar trends as with the test set. The confusion matrices of the Aliens Act with all sentences also show a similar trend, but the Other class ('O') is bigger and the performance of especially that class is higher for the rule-based method.

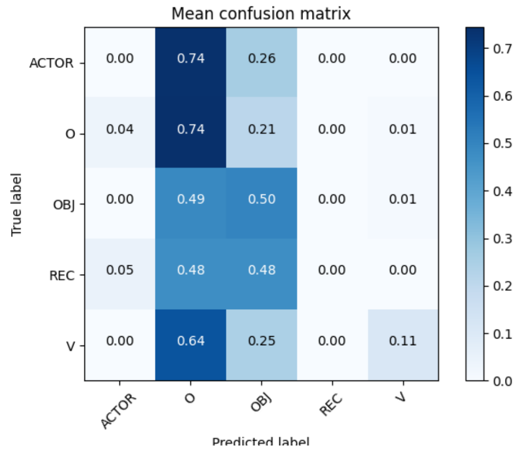

Figure 2: Mean Confusion Matrix (normalised) of the rule-based method on the test dataset

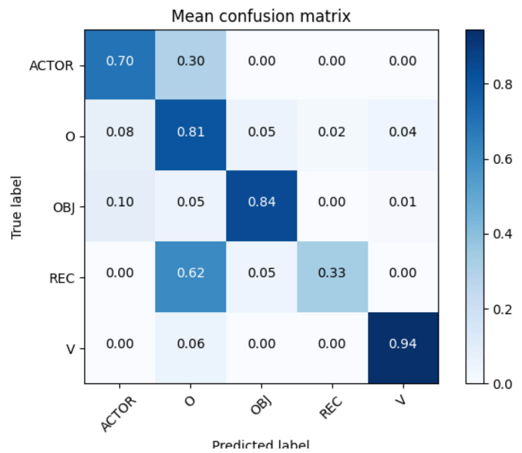

Figure 3: Mean Confusion Matrix (normalised) of the transformer-based method on the test dataset

5. Discussion

The results show that both on the test set and the Aliens Act the transformer-based method performed much better than the rule-based method. There are several explanations for the poor performance of the rule-based method, but the main one lies in the rules for recognising the different roles. First of all, for the rule-based method we did not design a rule to classify the recipient, hence recipients were not recognised. Second, the labelling of verbs is done according to a list of action verbs. If this list is not exhaustive, verbs will be missed. Since verbs were mislabeled with 'O' many times, this indicates that the list with action verbs should either be extended, or another rule should be added. Third, actors are rarely classified correctly in the rule-based method, and relatively often misclassified as the object. This indicates that the rule for recognising the actor is probably too narrow, and the rule for recognising the object too broad. This also shows in the results of classifying the object; all roles are, after the 'O' label, most often (mis)classified as the object.

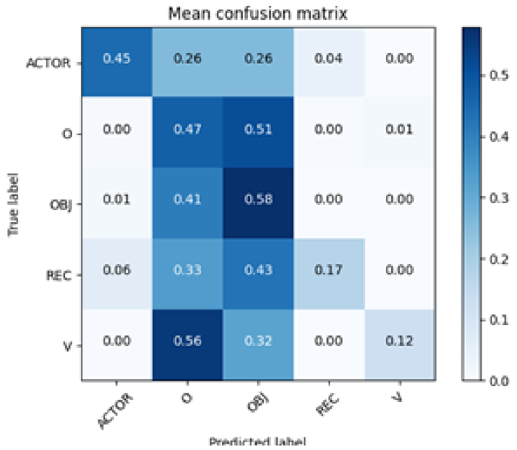

Figure 4: Mean Confusion Matrix (normalised) of the rule-based method on the Aliens Act

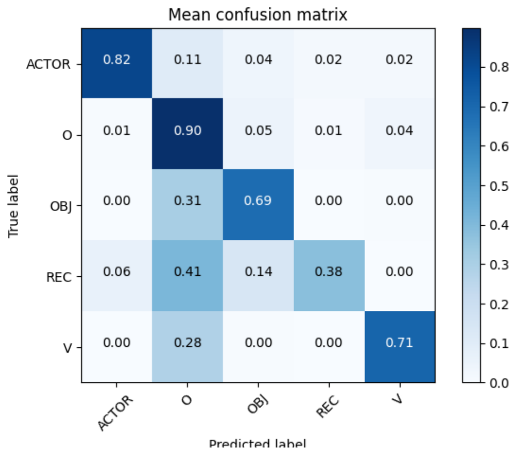

Figure 5: Mean Confusion Matrix (normalised) of the transformer-based method on the Aliens Act

Finally, we have to conclude that the performance of this method stands or falls with the quality of its rules. In the previous chapter, we compared this method with the transformer-based method and concluded that the last one performed better. However, the comparison is not completely fair since the performance of the rule-based method could be improved by elaborating its rules, thereby reducing the performance gap. de Maat et al. (2009) designed a more detailed rule-based approach which could be used to extend our approach in the future.

The transformer-based method showed a better overall performance with one exception: the mean accuracy score calculated on all (including sentences without actions) annotated sentences of the Aliens Act. This can be explained by how the transformer method is trained. The last layer of the model is fine-tuned on only positive sentences and therefore outputs at least a verb for most sentences. This entails that the transformer-based method will label verbs in sentences that do not contain acts at all (negative sentences). With respect to the transformer-based method it is interesting to note that it does not perform as good on classifying recipients as it does for the other labels. This might be explained by looking at our annotated training dataset. Most sentences contain a verb, actor, and object, but for many sentences there is no recipient. Since our annotated dataset is already very small, there are probably not enough recipient examples for our model to be trained properly. Finally, it might be interesting to compare the performance of our fine-tuned model to another model. For Dutch this is currently not possible since a model on semantic role labeling does not exist.

The results show that the transformer-based method is a promising approach of automatically extracting (the roles for) the Flint frames. The points stated above indicate that the method can be further improved by 1) annotating more, and more diverse, sentences for fine-tuning our model, 2) improving and elaborating on the rules of the rule-based method. The option of combining both methods is also worth exploring in the future.

6. Conclusion and Future Work

In this paper, we describe a first version of a method for interpreting legal texts by automatically translating legislation from semi-free texts into structured Flint frames. These frames are human- and machine-readable and offer a formal model enabling the unambiguous interpretation of such text by describing the norms in such texts as acts, facts and duties. In our method we preprocessed Dutch law documents to a structured format on sentence level. Then we labeled roles in these sentences based on two different approaches, a rule-based method and a transformer-based method. The results showed that, on both the test set generated from multiple laws and the separately annotated Aliens Act, the transformer-based method outperforms the rule-based method. The rule-based method lacked precision in its rules and therefore often misclassified the labels. The transformer-based method has a higher performance, and could even be improved further by fine-tuning the last layer on more annotated data. The results of the transformer-based approach show that this is a promising first step towards fully automatic extraction of act frames.

For future work it would be interesting to expand this method towards the extraction of other elements of the act frame, such as preconditions. The performance of the current transformer-based method could also be improved by training the model with more annotated data. Furthermore, it would be valuable to evaluate this method on English law texts, since there is a large variety of language models already available in English. In this study we looked at the different roles of an act frame, but we did not combine them automatically in an actual frame. Combining the rule-based approach and the transformer-based approach might be the key for fully automatic extraction of Flint frames.

Acknowledgments

The authors would like to thank the Governmental Advisory Board on Digital Government (OBDO) and Dutch Ministry of the Interior and Kingdom Relations for the financial support of our research. Furthermore we would like to thank our interns that annotated (Chris Slewe, Geanne Barkmeijer, Carlo Geertse), our colleague Ellen van Bergen who helped organising the annotations, the Norm Engineering project team, particularly Claudio Lazo and Georgios Stathis who manually created some act frames for the GDPR, and Jeroen Breteler and Daan Vos for proofreading.

7. Bibliographical References

Agnoloni, T., Francesconi, E., and Spinosa, P. (2007). xmlegeseditor: an opensource visual xml editor for supporting legal national standards. In *Proceedings of the V legislative XML workshop*, pages 239–251.

Biagioli, C., Francesconi, E., Passerini, A., Montemagni, S., and Soria, C. (2005). Automatic semantics extraction in law documents. In *Proceedings of the 10th International Conference on Artificial Intelligence and Law*, pages 133–140.

Bouma, G., Van Noord, G., and Malouf, R. (2001). Alpino: Wide-coverage computational analysis of dutch. *Language and Computers*, 37:45–59.

Brighi, R., Lesmo, L., Mazzei, A., Palmirani, M., and Radicioni, D. P. (2008). Towards semantic interpretation of legal modifications through deep syntactic analysis. In *Proceedings of The Twentieth First Annual Conference on Legal Knowledge and Information Systems*, volume 21, page 202.

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., and Androutsopoulos, I. (2020). Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.

de Maat, E. D. and van Engers, T. (2003). Mission impossible?: Automated norm analysis of legal texts. In *Legal Knowledge and Information Systems, Jurix 2003: the Sixteenth Annual Conference*.

de Maat, E., Winkels, R., and Engers, T. (2009). Making sense of legal texts. In Günther Grewendorf et al., editors, *Formal Linguistics and Law*, pages 225–256. De Gruyter Mouton.

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

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

Elwany, E., Moore, D., and Oberoi, G. (2019). Bert goes to law school: Quantifying the competitive advantage of access to large legal corpora in contract understanding. *arXiv preprint arXiv:1911.00473*.

Francesconi, E. (2006). The "norme in rete" project: Standards and tools for italian legislation. *International Journal of Legal Information*, 34(2):358–376.

He, L., Lee, K., Levy, O., and Zettlemoyer, L. (2018a). Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369, Melbourne, Australia. Association for Computational Linguistics.

He, S., Li, Z., Zhao, H., and Bai, H. (2018b). Syntax for semantic role labeling, to be, or not to be. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2061–2071, Melbourne, Australia. Association for Computational Linguistics.

Hoekstra, R. (2009). *Ontology Representation: Design patterns and ontologies that make sense*. IOS Press.

McCormick, C. and Ryan, N. (2019). Bert fine-tuning tutorial with pytorch.

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

Shi, P. and Lin, J. (2019). Simple bert models for relation extraction and semantic role labeling. *ArXiv*, abs/1904.05255.

Spinosa, P., Giardiello, G., Cherubini, M., Marchi, S., Venturi, G., and Montemagni, S. (2009). NLP-based metadata extraction for legal text consolidation. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 40–49.

Strubell, E., Verga, P., Andor, D., Weiss, D., and McCallum, A. (2018). Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Tamburini, F. (2012). The anita-lemmatiser: A tool for accurate lemmatisation of italian texts. In *International Workshop on Evaluation of Natural Language and Speech Tool for Italian*, pages 266–273. Springer.

van Doesburg, R. and van Engers, T. M. (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*, pages 27–37.

van Gog, R. and van Engers, T. M. (2001). Modeling legislation using natural language processing.

In *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat. No. 01CH37236)*, volume 1, pages 561–566. IEEE.

Winkels, R., Boer, A., de Maat, E., van Engers, T., Breebaart, M., and Melger, H. (2005). Constructing a semantic network for legal content. In *Proceedings of the 10th International Conference on Artificial Intelligence and Law*, pages 125–132.

Appendix

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

Table 4: Example of a manually created act frame from the GDPR, where <<>> is used to denote an act, <> to denote a duty and [] to denote a fact.