PUNISHMENT EXTRACTION FROM DUTCH CRIMINAL CASES IN COURTS OF FIRST INSTANCE

Edwin Wenink / Johan Kwisthout / Tom van Engers

Edwin Wenink Data Scientist, Independent Researcher
edwinwenink@hotmail.com; https://www.edwinwenink.xyz

Johan Kwisthout Associate Professor, Radboud University Nijmegen, Donders Institute for Brain, Cognition and Behaviour.
j.kwisthout@donders.ru.nl; https://www.socsci.ru.nl/johank/

Tom van Engers Professor, University of Amsterdam and TNO / Leibniz Institute
vanengers@uva.nl; http://www.leibnizcenter.org/

Keywords: ***punishment extraction, Dutch criminal law, case law, regular expressions***

Abstract: *This paper explores the merits of a pattern- and rule-based approach for the automated extraction of punishments from Dutch criminal cases in courts of first instance. Automated extraction of case outcomes leverages the increasing amount of information becoming available through digital technologies and aids the creation of big data sets for work in legal informatics and AI & Law. This work addresses domain-specific challenges, in particular that Dutch criminal case decisions may impose a single combined sentence for multiple facts or impose multiple sentences in the same decision. Manual evaluation of the developed method shows that the use of interpretable methods is a viable approach in the legal domain.*

1. Introduction

A main promise of AI, machine learning and data science in the legal domain is to automatically identify legal factors or even arguments that contribute to a particular case outcome [Falakmasir and Ashley 2017, Westermann, et al. 2019, Wyner, et al. 2010]. This could also be an undesirable outcome in order to prepare against possible (counter)arguments from the opposing party [Shulayeva, Siddharthan and Wyner 2017, 108]. This paper presents the first part of our study aimed at understanding the contribution of case factors and circumstances to the composed case decision. In this paper we focus on making those case outcome labels available by automatic extraction of punishments imposed in Dutch criminal case law, while the second paper is focused on the analysis of the relationships between the constituent components of the decisions and their relationships with case factors. Related work typically deals with decisions on a particular topic, such as landlord-tenant [Westermann, et al. 2019] or eviction decisions [Medvedeva, et al. 2021]. This project is broadly scoped to criminal cases from courts of first instance, but is otherwise topic agnostic.

Rather than opting for a widely used machine learning (ML) approach working on manually labelled cases, we have chosen to not neglect the expert knowledge that is rather widely available in this domain and reflected in the "tradition" of writing court decisions. Some wordings and chosen phrases in case law are frequently used patterns with specific semantics meaningful to legal professionals, so why not explicitly address them. For this reason, we use regular expressions to capture typical legal language expressions. This allows us to connect those explicit patterns to case factors and circumstances (the focus of the second paper) without suffering from the typical issues of ML, including lack of explainability and hiding away domain knowledge in the data preparation stage of ML-projects. In short, rather than using ML to mimic human-labelling of cases, which seems to be the mainstream in legal AI these days, we study to what extent we can construct an explicit knowledge model of the structure and relationships of relevant concepts expressed in natural language in case law.

The availability of case outcome labels is valuable for downstream applications such as decision-support systems that use case-based reasoning. Case-based reasoning "is a meta-level of argument concerning the cases themselves" [Wyner, et al. 2010, 72] where one looks to comparable prior cases when pleading the case at hand or predicting its outcome. This is particularly prevalent in common law where past decisions serve as precedents for the current case and where, as a rule of thumb, comparable cases have similar outcomes. Even though the case base does not directly set law-giving precedents in the Dutch *civil law* tradition, case law still has a similar function as in common law. The *interpretation* of previous cases guides how norms and laws are applied in practice and thus are part of the law-giving role of judges [Van Opijnen 2014, 410–411]. The study of case-based reasoning should thus not be restricted to common law only. This work aids studies in the civil law tradition by making case outcome labels available for Dutch criminal law.

Moreover, both legal traditions face similar challenges as they have to deal with a case base of increasing size that requires extensive training to interpret and navigate [Wyner, et al. 2010, 61]. Manual analysis of cases is costly, but also does not scale well with the increasing amount of information becoming available through digital technologies. This form of information overload can be alleviated by the automatic extraction of useful information.

We address a domain-specific challenge of Dutch criminal case decisions, namely that they may impose a single combined sentence for multiple facts, as well as impose multiple sentences in the same decision. Moreover, in order to live up to its potential, any form of automated information extraction needs to meet the high standards of legal practice and must be interpretable, because inaccurate information may have large consequences in the high-risk legal domain. With these challenges in mind, this paper explores the merits of a pattern- and rule-based approach for automated punishment extraction. Manual evaluation of the developed method resulted in promising performance, which shows that the use of interpretable methods is a viable approach in the legal domain.[1]

2. Background

Dutch criminal law (Art. 9 Wetboek van Strafrecht[2]) distinguishes four main types of punishment ("hoofdstraffen"): prison sentence ("gevangenisstraf"); custody ("hechtenis"); community service ("taakstraf"); fine ("geldboete"). These punishments are largely self-explanatory. The difference between a prison sentence and custody is somewhat subtle. They are similar in practice because they are both custodial sentences, but they are theoretically distinct. Custody is a punishment for (severe) offences, whereas prison sentence is a punishment for crimes. Custody in the Netherlands typically has a maximum length of one year, whereas a prison sentence has a maximum duration of 30 years.[3] Custody as a main punishment is post-trial and is thus not the same as detention while the trial is pending.

There is a set of additional secondary punishments a judge can impose (such as making the case public without anonymization) as well additional measures to take (such as placement in a psychiatric hospital). For feasibility, we initially focus on the four main types of punishments. However, one particular measure called TBS (placement under hospital order) will be treated as an exception due to its severity. TBS can be imposed for severe crimes that have a minimum prison sentence of four years and some other specific crimes in cases

[1] The often cited trade-off between performance and interpretability does not necessarily hold in general [Arrieta, et al. 2020] and some even call this perceived trade-off a harmful "myth" that discourages data scientists from developing interpretable machine learning models [Rudin 2019]. In some problem domains simple transparent models can achieve very good performance and some research indicates that maybe only in exceptional cases opaque models perform better such that the cost of losing explainability is worth it [Schwartzenberg, van Engers and Li 2020, 269].

[2] https://wetten.overheid.nl/BWBR0001854/2022-07-01.

[3] In Dutch criminal law there is however also a controversial prison sentence for life, which is only imposed in very exceptional cases. In this project we parse the length of prison sentences, but it is unclear how to quantify the length of a lifelong sentence. We therefore consider this exceptional sentence as out of scope for this project, which has no serious impact on our results due to its extreme rarity.

where the suspect suffered from a mental disease or mental disturbance at the time of the crime. The goal of this type of punishment is additional psychological treatment to avoid recidivism. However, what makes this particular punishment potentially much heavier than a prison sentence, despite formally being a "measure" and not a "main punishment," is that the patient needs to pass a psychiatric evaluation before being a candidate for release. TBS is initially imposed for a minimum of two years but without a predefined maximum duration, and is evaluated every two years. If after six years the evaluation is still not positive, the detainee will be marked as a long-stay patient. At the extreme, this can lead to a lifelong detainment, despite not formally being a life sentence. We therefore also include TBS as a punishment, despite not formally being one.

2.1. Defining criminal case outcomes

In order to label cases with their outcome, it is important to clearly define what we mean with "case outcome." Al-Abdulkarim et al. [Al-Abdulkarim, Atkinson and Bench-Capon 2016, 3] define "verdict" as a "binary decision such as guilty or innocent and recognize that, when appropriate, sentence and damages remain to be determined." They also note that most approaches in AI and Law "represent legal concepts only as Booleans. For example factors are considered present or absent and values are promoted or demoted, whereas we believe that differing degrees of presence and absence (...) need to be recognised." [Al-Abdulkarim, Atkinson and Bench-Capon 2016, 3]. Related work on Dutch eviction cases also only uses binary verdicts [Medvedeva, et al. 2021, 4]. Following this line of reasoning, it is preferable to define the case outcome as the assigned sentence (i.e. one of the *types* of punishments we identified) including its "height" (length or sum of money). The sentence height contains information that a Boolean verdict does not have, because we expect higher sentences to be assigned for more severe crimes and offenses.

Extracting both the punishment type and the precise height of a punishment is considerably more difficult than extracting a Boolean verdict and also adds complexity to machine learning and reasoning approaches using these labels. Another source of complexity is that cases in Dutch criminal law often consider multiple charged facts that each may call for a different punishment, but that the final ruling by the judge passes sentences that addresses all charges at once. In many cases, this means that two different types of punishment co-occur in the same ruling, e.g. a prison sentence alongside a fine. But in case two charged facts call for the same punishment type, for example a prison sentence, a *single* prison sentence that addresses both charged facts is imposed if the defendant is found guilty. This is a notable difference with for example the U.S. legal system, where it is possible to receive N cumulative life sentences if N crimes with a life sentence as punishment are committed. This project aims to be sensitive to these nuances in Dutch criminal law – and is innovative in this regard – by modelling the case outcome as a vector or tuple, where each dimension corresponds to a punishment *type* and the value for each dimension corresponds to the *height* of that respective punishment.

The interpretation of "case outcome" in this project is thus the total collection of sentences, including their height, imposed on the defendant in the case by decision of the judge(s). The sentence heights are real-valued variables, with two exceptions. Firstly, we have already discussed that TBS is often assigned without a clear maximum duration and hence we record this punishment type as a binary variable. Secondly, we also record whether acquittal is detected, but acquittal does not have a "height." We do observe that acquittal often occurs multiple times in the same case, but this is because multiple (possibly disconnected) charges are treated in the same case and not because the suspect is twice as innocent. It may also be the case that there is a primary charge for a particular fact, for example murder, but also a subsidiary charge in case there is not sufficient proof for the primary charge, such as manslaughter. A judge is required to rule on all charges, so even for a single fact a final decision may for example contain acquittal on the count of murder, but the imposition of a prison sentence for manslaughter. Moreover, even if acquittal is detected without any main punishment, we still cannot say with certainty that the defendant is not guilty, because we limited the scope to the main punishments and TBS and thus do not match the myriad of all measures and secondary punishments. If we

detect acquittal without any other imposed main punishment, we can strictly speaking only say the defendant is acquitted of all charged facts for which a main punishment or TBS would be required.

3. Methods

3.1. Data set

A collection of Dutch cases called *Open Data van de Rechtspraak* is publicly available.[4] Cases are identified by the European Case Law Identifier (ECLI). We queried court rulings ("uitspraak") rather than juridical reflections from higher courts ("conclusie"). We limited ourselves to cases from criminal law, but not one specific subfield of criminal law. We only queried cases from 2021 to control the data set size.

Most of the cases that have been added after the introduction of the ECLI format in 2010 have an XML representation that distinguishes case sections and typically labels the sections containing the case decision as such. Cases lacking this richer XML representation were discarded. For each section, the section title was parsed and recorded separately. Section titles are free text, but nevertheless the legal clerks writing up the cases use common formulations for describing the content of a section. This means that these section titles can be interpreted as weak labels and used to label sections for their juridical significance in the overall case text. We designed a rule-based labelling scheme to produce more section labels based on the common structure of Dutch criminal law cases and section titles, but in this project we only used the sections containing decisions. If a case decision was not yet labelled by rechtspraak.nl, we used a back-up rule: if "beslissing" or "vrijspraak" occurs in the section title, and we have not registered indications of other section types, then we label the section as a case decision. Finally, we parsed metadata on the procedure type to only retain the cases from courts of first instance where a punishment is imposed on a suspect.

3.2. Detecting case sentences using regular expressions and rules

Manual inspection of case transcriptions showed that the transcriptions of a judge's ruling tend to use common phrasings. We exploited this regularity by writing a regular expression that parses both the type of punishment and the measure of punishment. This regularity of juridical language also makes approaches like context-free grammars quite successful, even for detecting argumentation structures [Wyner, et al. 2010, 10].

The four main punishments of Dutch criminal law all share a common feature, namely that the judge is always required to specify the length or height of the punishment. The duration of a punishment is practically always mentioned with a digit alongside the number fully written out, and comes *after* a mention of the type of punishment. This is different for TBS, which does not have a preset duration, and for acquittal, which is a binary decision. Rather than designing multiple regular expressions for different punishments and different situations, we designed a single regular expression to match all four main punishments. So rather than only matching exactly the information we need (so that we can directly use all matches), we captured all potentially relevant information and post-process it with a set of rules. The chosen approach avoids having to do extra passes over the input text, avoids capturing the same information twice, and allows for more flexibility because the post-processing rules are easier to adjust than the regular expression itself. The length of a sentence or the amount of a fine is captured as a free form description in natural language, but is further processed and normalized as the number of days or the amount of euros. Two additional regular expressions are used for TBS and acquittal, because they do have a different linguistic representation without a specification of the punish-

[4] https://www.rechtspraak.nl/Uitspraken/paginas/open-data.aspx.

ment height. The supplementary materials[5] contain a detailed explanation of the used regular expressions and of how the parsing procedure handles a variety of scenarios, as well as the code implementation.

We limit ourselves here to some brief descriptions of interesting scenarios. For example, we regularly have a compounded description of prison sentences[6] which requires matching two time units and a conversion into days in order to sum them. However, in several scenarios we match two units of time that we do *not* want to sum up together, for example when the second part indicates a conditional part of the sentence or a probation:

> **Match prison sentence**: gevangenisstraf van 12 (twaalf) maanden, waarvan 6 (zes) maanden voorwaardelijk met een proeftijd van twee jaar
>
> **Warning:** Second part of the match is conditional. This part is excluded.
>
> **Output:** {'TBS': 0, 'prison sentence': 365, 'custody': 0, 'community service': 0, 'fine': 0, 'acquittal': 0

We filter out measures where possible for the sake of consistency, because we have limited our scope to main punishments. In particular fines (formally a *punishment*) are easily confused with monetary *measures* and are a source of false positives. For fines and community service we typically have custody as a subsidiary punishment in case the main punishment is not executed. When this is not recognized this can lead to very wrong results, such as community service in the order of 30 days, whereas the legal maximum is 10 days. Consider the following example:

> **Match community service:** taakstraf bestaande uit het verrichten van onbetaalde arbeid voor de duur van 60 (zestig) uren, subsidiair 30 dagen hechtenis
>
> **Warning:** Subsidiary punishment detected. This part is excluded.
>
> **Output:** {'TBS': 0, 'prison sentence': 0, 'custody': 0, 'community service': 3, 'fine': 0, 'acquittal': 0}

There are many other interesting scenarios that are handled by the parser. Often a suspect will have spent time in custody awaiting trial and this time will be reduced from whatever sentence is ultimately imposed. We want to keep the original sentence that is imposed for the crime, but not add (nor subtract) the time already spent in custody because this is an empirical contingency. It also occurs regularly that a (part of a) sentence is mentioned precisely because it is *not* executed. Acquittal on some fact is relatively easy to match, but in some cases we find an indication of acquittal that is a formal statement that excludes further prosecution on the same facts. This indicates the "ne bis in idem" principle (cf. "no double jeopardy"), but should not be considered acquittal of the charged facts:

> **Match acquittal:** spreekt verdachte vrij van wat meer of anders is ten laste gelegd
>
> **Warning:** "ne bis in idem" detected. Skipped.
>
> **Output:** {'TBS': 0, 'prison sentence': 0, 'custody': 0, 'community service': 0, 'fine': 0, 'acquittal': 0}

3.3. Selection bias

The Public Prosecution Service (*Openbaar Ministerie*) in the Netherlands decides which cases appear in front of a judge. Because this body has limited resources, the decision to let a case appear in court is partially based on the probability of reaching a conviction. This implies that we can expect that the majority of the cases that appear in court leads to a conviction. Petty crimes are more likely to be handled outside of the courtroom and will thus appear less in the data set. Because more severe crimes also have heavier punishments, we also expect a selection bias towards the heavier sentences such as imprisonment. A high selection bias towards a particular punishment type does not necessarily imply that this punishment type is more frequent in absolute

[5] Supplementary materials: https://github.com/EdwinWenink/dutch-criminal-law-punishment-extraction/.

[6] A duration of 2.5 year is written as "2 (two) years and 6 (six) months."

terms. For example, we can expect a very high selection bias towards TBS cases, but we do not expect them to outnumber prison sentences because TBS cases are rare.

There is a second source of selection bias. Not all cases that appear in court are also publicized in *Open Data van de Rechtspraak*. The selection of cases is important for countering information overload: one wants to present the more interesting cases that are likely more valuable for legal professionals, rather than a plenitude of cases that are all nearly identical.[7] Data collections thus do not always aim to be representative in a strictly statistical sense, because the average case is not a new landmark in jurisprudence.

3.4. From sentences to labels

For typical downstream machine learning tasks such as classification, it is common to have a single target label. We have established, however, that a single case often discusses multiple facts and charges and consequently may contain multiple sentences in the case decision. We may either apply machine learning approaches on these label vectors as a multi-output problem, or define an approach to convert the label vectors to a single label. Multi-output approaches typically decompose the multi-output problem into multiple single-output problems, but this assumes independence of the punishment vector dimensions. We therefore provide an analysis of the co-occurrence of punishment labels in section 4.2. This additionally provides insight into the problem domain.

Note that it is not a good idea to just use the most severe punishment as the target label in downstream applications. Firstly, this discards information on subsidiary punishments that may be relevant for some applications. Secondly, by assigning a higher priority to more severe punishments, the effect of the selection bias towards heavier sentences is amplified. We have instead opted for a clustering approach in order to find common patterns in punishment vectors without discarding information on co-occurring sentences. The first results suggest that these clusters capture meaningful patterns such that the cluster labels may be used in further analyses or as the target label in machine learning applications. These results will be presented in a follow-up paper.

4. Results

4.1. Performance metrics

The regular expressions used for detecting punishments in Dutch case decisions have been developed on a subset of cases from late 2020 and manually validated on a test set of 35 randomly selected decisions from 2021.[8] These 35 cases contained 50 sentences, of which 45 were recognized as true positives (TP) and 5 as false negatives (FN). There were 57 sentences that superficially resembled punishments and were matched by the regular expressions. Of these, 52 were correctly recognized by the rule-based classifier as true negatives (TN), i.e. not counted as punishments, but 5 were nevertheless counted as false positives (FP). Note that in order to count as correct both the punishment *type* (6 types) and *height* need to be parsed correctly. This results in an F1-score of 0.90, with equal precision and recall.

This decent F1-score reaffirms that court transcriptions use conventional writing patterns that we can exploit. Furthermore, a failure analysis (see supplementary materials) showed that the mistakes (FP and FN) mainly follow a limited set of patterns that are currently not caught by the regular expressions, but could be in the next iteration of this work to further improve the F1-score. We consider an extension to the chosen approach to be feasible because there is a limited amount of punishments and measures specified in criminal law, and so far we have observed that each of these legal concepts also have a limited set of linguistic representations.

[7] https://www.rechtspraak.nl/Uitspraken/Paginas/Selectiecriteria.aspx.

[8] The ECLIs and manual validation results can be found in the supplementary materials.

4.2. Punishment vector statistics

We expect that case decisions will on average contain more than one punishment, either because multiple charges are discussed in the same court hearing or because there are subsidiary charges for the same facts. To asses this, we computed the label cardinality of the punishment vectors as the average number of punishment types per case decision. For our data set with 2945 decision sections from the year 2021, this resulted in a label cardinality of 1.44, so on average cases indeed have more than one *type* of punishment. Moreover, there can be more instances of the same type in a single decision.

We computed the co-occurrence matrix over the punishment vectors to gain more insight into the co-occurrence of punishments. To assess how frequent label *i* and *j* co-occur relative to their overall frequency, we furthermore normalized the co-occurrence counts by dividing by the total occurrences of *i or j* (Jaccard Index). Figures 1 and 2 show the co-occurrence matrices for our data set of cases from 2021.

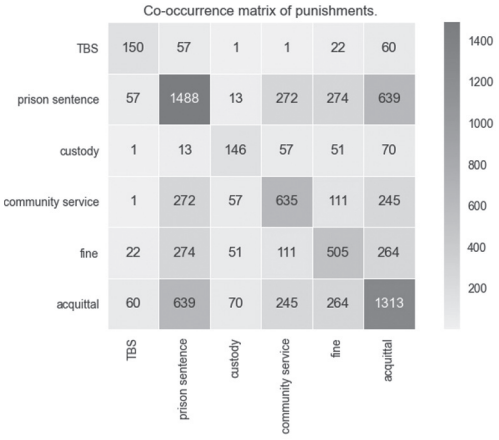

Figure 1: Punishment co-occurrence matrix showing absolute counts.

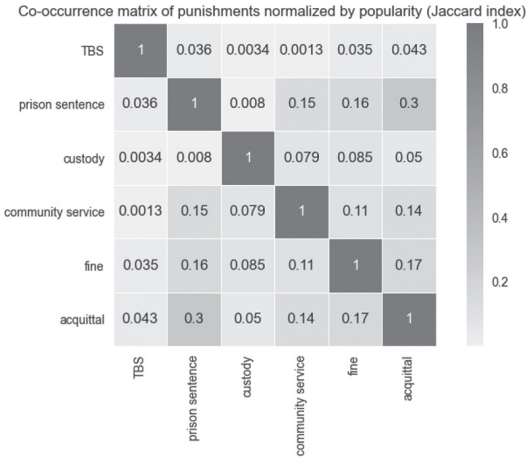

Figure 2: Punishment co-occurrence matrix normalized by popularity (Jaccard index).

Figure 1 shows that some pairs occur practically never, such as TBS-custody (N=1), TBS-community service (N=1), and prison-custody (N=13), but this is partly explained by TBS and custody being rare punishments overall (N=150; N=146). The Jaccard Index better expresses how notable a co-occurrence is relative to the overall punishment frequency. Figure 2 shows relatively high Jaccard indices for prison sentence-acquittal (.3) and community service-acquittal (.14). Given our hypothesis that there is a bias towards conviction, 1313 cases with acquittal seems high on a total of 2945 cases, but note that acquittal co-occurs with prison sentences 639 times, with community service 245 times and with fines 264 times. Further inspection shows that there are only 395 cases where acquittal is detected without other main punishments. Co-occurrences with acquittal thus make a relatively large contribution to the label cardinality.

We also assess whether the *heights* of co-occurring punishments correlate. We do not have normally distributed data because the punishment distributions are heavily right-skewed, meaning that very heavy punishments are much more uncommon than moderate punishments. We also have outliers that are nevertheless legitimate data points given our domain knowledge. For example, fines can range from €3 to €900.000. We address this by computing Pearson's correlation over the *rank* of the data (i.e. Spearman's correlation), which only assumes monotonicity and is less sensitive to outliers. We find the following Spearman ρ coefficients for co-occurring sentences: prison sentence and custody (ρ=.18, p=.55); prison sentence and community service (ρ=.34, p=1.43e-08); prison sentence and fine (ρ=.32, p=6.62e-08); custody and community service

(ρ=−0.0059, p=.97); custody and fine (ρ=.41, p=2.63e-03); community service and fine (ρ=.25, p=7.39e-03).[9] We thus find multiple significant positive correlations of moderate strength, suggesting that *when* these punishments co-occur, they tend to increase in height together. This may be because the underlying charge in the case is more severe, but we will have to interpret this finding in light of our domain knowledge. The insignificant results are not surprising because the respective pairs have very few co-occurrences.

5. Discussion and Conclusion

This work on punishment extraction is an interesting starting point for follow-up research, particularly because there is a lack of openly available annotated data sets in the Dutch language. Because the annotation process is fully automatic, it becomes feasible to build larger data sets for use in data science. For example, we could test to what extent public sentiment motivates judges to assign higher punishments for certain types of crime if that crime incited public rage due to some in-world event. Such annotated data set could also enable the temporal analysis of case outcomes and potentially show outliers that start a new trend in the case base. In a follow-up paper, we will present the analysis of case factors and circumstances, structures and relationships to the punishments that we now can automatically annotate in a much larger corpus. As we stated before we deliberately opted for a different approach than the mainstream ML typical for legal AI these days. The main purpose of the part described in this paper was not to predict case outcomes, but rather to create an explicit and explainable representation of typical legal patterns present in case law that legal professionals are familiar with and that have specific semantics.

Manual evaluation showed that the extraction of concurrent punishments from case decisions using regular expressions and rules was surprisingly effective. This first of all confirms the preconception that juridical language is relatively formal for a natural language and tends to follow conventions in the way information is conveyed. This secondly shows that, despite the massive contemporary interest for machine learning, we should not disregard the utility of regular expressions, context free grammars, and rule-based approaches, specifically in the legal domain. An obvious downside is that these methods are not robust to the inherent variability in natural language, which may be as trivial as a spelling mistake. However, this downside arguably ways much heavier in other domains than in law, because in the legal domain much care is taken to write down information *verbatim* according to concepts *grounded* in the law, without ambiguity, and without spelling mistakes. Advanced ML approaches may deal better with linguistic variation, but this comes at a cost. ML approaches are data hungry and require a large tagging effort in the data preparation phase, where spelling mistakes and uncommon phrasings would have to be tagged too. Moreover, the domain knowledge that is used for tagging is abstracted away in the data set and becomes implicit. But the more serious problem in the high-risk legal domain is the loss of interpretability. For every detected punishment it becomes a research problem in itself to explain *why* the model detected a punishment, but more importantly it becomes non-trivial to explain why the model did *not* detect a punishment or why it detected the wrong one. The introduction of this uncertainty in the labelling process itself limits the practical applicability of the labels in downstream applications. This problem does not occur in our chosen approach because it specifies an unambiguous mapping between legal concepts and linguistic patterns. Importantly, this mapping is *explicit*, which means that legal experts can inspect it, discuss it, and alter it if necessary. Apart from the initial manual effort to build a representation of legal concepts such as criminal punishments, the performance of our approach is not dependent on the size of the data set.

It remains an open question what the co-occurrences of particular punishments signify. In Dutch criminal law it is possible that several unrelated facts (unrelated apart from having the same perpetrator) are treated in the same trial. Ipso facto, there is no guarantee that the co-occurrence of two or more punishments relate to the

[9] TBS and acquittal are not reported on here, because we cannot compute rank correlation with a constant array of ones.

same facts. This in turn is an a priori argument against the hypothesis that the severity of an underlying fact may explain the positive correlation of co-occurring sentences, since these sentences may or may not relate to the same fact. Humans can interpret and understand the relation between punishments and facts, but this is a challenge for a purely data-driven analysis. For one, it should be taken into account that criminal law places constraints on which punishment is suitable for which fact. For example, TBS is only asked for by the prosecution and imposed by the judge for very severe crimes, for which custody or community service is simply not a possible punishment according to Dutch criminal law. Therefore, we can be sure that these punishments do *not* relate to the same fact complex.

Despite having no guarantee that the charged facts in a case are related, the charged facts are related *by construction* in the case of subsidiary charges. It could for example be the case that a suspect is charged with murder ("moord"), with manslaughter ("doodslag") or involuntary manslaughter ("dood door schuld") as a subsidiary charged fact. These charged facts are related but also mutually exclusive, because if murder is proven, the suspect will not additionally be prosecuted for the subsidiary charges. So generally speaking subsidiary facts do not explain the co-occurrence of punishments, with one exception: acquittal. This explains to a large degree why, somewhat surprisingly, acquittal co-occurs in so many cases, even after filtering out the *ne bis in idem* construction. Interestingly, we thus observe that the high occurrence of acquittal in case decisions does not contradict our hypothesis that the vast majority of cases will lead to a conviction.

Acknowledgement This research is partially supported by the appl.ai program of TNO.

6. References

al-abdulkarim, latifa/atkinson katie/bench-capon, trevor, Statement Types in Legal Argument. In: Bex, Floris/Villata, Serena, Legal Knowledge and Information Systems, IOS Press, 2016, p. 3–12.

Arrieta, Alejandro Barredo, et al., Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Information Fusion, 2020, 58, p. 82–115.

Falakmasir, Mohammad H./Ashley, Kevin D., Utilizing Vector Space Models for Identifying Legal Factors from Text, Legal Knowledge and Information Systems, 2017, 302, p. 193–192.

Medvedeva, Masha/Dam, Thijmen/Wieling, Martijn/Vols, Michel, Automatically identifying eviction cases and outcomes within case law of Dutch Courts of First Instance. In: Schweighofer, Erich, Legal Knowledge and Information Systems, IOS Press, 2021, p. 13–22.

Rudin, Cynthia, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature Machine Intelligence, 2019, 1, p. 206–215.

Schwartzenberg, Carel/Van Engers, Tom/Li, Yuan, The fidelity of global surrogates in interpretable Machine Learning. In: Cao, Lu/Kosters,Walter/Lijffijt, Jefrey, Proceedings from BNAIC/BeneLearn 2020, 2020, p. 269–283.

Shulayeva, Olga/Siddharthanm, Advaith/Wyner, Adam, Recognizing cited facts and principles in legal judgements, Artificial Intelligence and Law, 2017, 25, p. 107–126.

Van Opijnen, Marc, Op en in het web: Hoe de toegankelijkheid van rechterlijke uitspraken kan worden verbeterd. Ph.D. dissertation, Leibniz Center for Law, 2014.

Westermann, Hannes/Walker, Vern R./Ashley, Kevin D./Benyekhlef, Karim, Using Factors to Predict and Analyze Landlord-Tenant Decisions to Increase Access to Justice. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law, ACM, New York, 2019, p. 133–142.

Wyner, Adam/Mochales-Palau, Raquel\Moens, Marie-Francine/Milward, David, Approaches to Text Mining Arguments from Legal Cases. In: Francesconi, Enrico et al., *Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language*, Springer, Berlin, 2010, p. 60–79.