Supplementary Materials

Edwin Wenink

September 2022

Contents

1 Regular expression for detecting case sentences

Albeit being natural language in free form, juridical lingo tends to use relatively standardized formulations from common situations. This also holds the final verdict as summarized at the end of a case transcription. It is therefore feasible to extract information about the verdict in a structured manner using regular expressions. This appendix explains the used regular expressions, how they are processed, and discusses their limitations.

1.1 Main punishments

Dutch law distinguishes four main types of punishment: prison sentence, custody, community service and a fine. These all share a common feature, namely that the judge is always required to specify the length or height of the punishment. This is different for TBS, which does not have a pre-set duration, and for acquittal, which is a binary decision. Rather than designing multiple regular expressions for different punishments and different situations, I designed a single regular expression to match all of them. This implies a shift in mindset from only matching exactly the information we need (so that we can directly use all matches), to capturing all potentially relevant information and then defining a rule-based classifier on the captured information. The latter approach avoids having to do extra repeated passes over the input text, avoids capturing the same information twice, and allows for more flexibility because the rule-based classifier is easier to adjust than the regular expression itself. The complete pattern used for the main punishments is:

```
(?i)(?:(?P<modifier1>voorwaardelijk|proeftijd|niet|vervangend|ind
ien|mindering|maatregel)[^\n\r;.]{0,100})?\b(?P<straf>gevangenis|
gevangenisstraf|jeugddetentie|detentie|hechtenis|taakstraf|werkst
```

```
raf|leerstraf|geldboete|vordering(?!\stenuitvoerlegging)(?!\stot\
stenuitvoerlegging)|betaling)\b(?P<test1>[^\n\r;.]{0,85}?)(?P<num
mer1>(?<!feit␣)\d+(?!\s?\]))(?P<test2>[^0-9\n\r;.]{0,30}?)(?P<een
heid1>jaar|jaren|maanden|maand|week|weken|dag|dagen|uur|uren|euro
|,[-\d=]{1,2}|(?:\/|-|:)?[\d.]+(?!\s?\])(?:,[-\d=]{1,2})?)(?P<nie
ttest1>[^0-9\n\r;.]{0,15})(?:(?P<nummer2>(?<!feit␣)\d+(?!\s?\]))[
^0-9\n\r;.]{0,30}?(?:\s(?P<eenheid2>jaar|jaren|maanden|maand|week
|weken|dagen|dag|uur|uren)))?(?:(?P<niettest2>[^\n\r;.]{0,150})(?
P<modifier2>voorwaardelijk|proeftijd|niet|vervangend|indien|hecht
enis|wederrechtelijk))?
```

This regular expression is long and quite unreadable (as most regular expressions are), but it does have some repeated subcomponents that we will explain part by part here. The matching is case insensitive, indicated by `(?i)`. The most relevant components are captured as named groups, indicated with the `(?P<name>)` syntax. This syntax is specific to the native Python `re` module.

Main punishment `\b(?P<straf>gevangenis|gevangenisstraf|jeugddet entie|detentie|hechtenis|taakstraf|werkstraf|leerstraf|geldboete| vordering(?!\stenuitvoerlegging)(?!\stot\stenuitvoerlegging)|beta ling)\b`

The most important component is the indication of the type of punishment. Some synonyms are included that are mapped to the four formal names in the downstream labelling procedure. For example, "werkstraf" and "taakstraf" both indicate community service. The word "detentie" (detention) is ambiguous and could both mean a prison sentence and custody. We treat it as a synonym for custody ("hechtenis"), because we observe in the text data that this word is typically used for sentences with less than a year of detainment. The word "vordering" often indicates a monetary claim and thus relates to a fine. However, there are of course other types of claims. A very common one is a claim to execute a conditional punishment of which the conditions are violated ("vordering (tot) tenuitvoerlegging straf"). This common case is excluded by doing a negative lookahead on "tenuitvoerlegging" and "tot tenuitvoerlegging." We include a zero-width word boundary assertion e.g. to exclude matching "vordering" as in: "het Wetboek van Strafvordering ten hoogste kan worden gevorderd op 3 jaren" (ECLI:NL:RBLIM:2020:9692).

Punishment height `(?P<test1>[^\n\r;.]{0,85}?)(?P<nummer1>(?<!fe it␣)\d+(?!\s?\]))`

We then want to know the length or height of the punishment, which is either an amount of time (in hours, days, months, or years) or money (in euros). Monetary sums are always written out as digits. The length of the other sentences are almost without exception also indicated with digits in the following canonical form: "veroordeelt de verdachte tot een gevangenisstraf voor de duur van 2 (twee) jaar." Of course, digits may occur for a variety of reasons, so we need to ensure that a matching digit indicates the appropriate measurement. This is done in several ways.

First of all, the digit should be related to the main type of punishment and

therefore we require the digit to occur in a particular window after the indication of the punishment. We allow a break of maximum 85 characters. This window size has been empirically determined by evaluating test cases. Moreover, we require the digit to occur in the same sentence and break the match when a new sentence is detected (I refer to the regex parts such as `[^\n\r;.]` as "connectors"). Semicolons are heavily used in enumerations in the case transcriptions and are also treated as a line separator. This approach makes all regular expressions also work when all newlines are stripped from the case text.

Secondly, the matched digits are always coupled with the detection of the unit of measurement (see below for more details). If we find a prison sentence but the amount is expressed in euros, then we know the matched digits do not express the length of the prison sentence and we reject the match in the downstream logic. Likewise, if we find a fine but the digit expresses an amount of time, we reject the match. These situation occur because when a fine is imposed, there is typically an expression of a replacement punishment in case the fine is not paid. Examples are: "hechtenis heeft doorgebracht naar rato van 50 euro per dag" and "betaling en verhaal te vervangen door 100 dagen gijzeling." The specification of such a replacement punishment is more or less a formality and the height of the replacement punishment is coupled to the height of the original punishment. Since we are interested in detecting the main sentence, we can safely discard these matches.

Thirdly, we have to deal with some edge cases. All case transcriptions are anonymized, as follows: "Wijst de vordering van de benadeelde partij [slachtoffer 1] toe tot een bedrag van € 1.314,28 (duizend driehonderdveertien euro en achtentwintig cent)." The 1 in "[slachtoffer 1]" prevents a match of the amount of euros. Because this is a very common scenario that occurs in practically each case transcription, we do a negative lookahead and only match digits that are not succeeded by the closing bracket ']'. In a sentence like "betaling aan de benadeelde partij [Slachtoffer 3] (feit 9) van € 5.226,53" we also want to exclude the number 9 from the fine. This is done with a negative lookbehind for a preceding "feit". Digits also frequently occur in dates and in case identifiers (e.g. the "parketnummer"). See below how they are handled.

Unit of measurement `(?P<test2>[^0-9\n\r;.]{0,30}?)(?P<eenheid1>jaar|jaren|maanden|maand|week|weken|dag|dagen|uur|uren|euro|,[-\d=]{1,2}|(?:\/|-|:)?[\d.]+(?!\s?\])(?:,[-\d=]{1,2})?)`

Once we know the amount, we need to know the unit of measurement (hours, days, months, years, or euros). Again, we allow a limited window of characters between the number and the unit, e.g. as in "duur van 2 (twee) jaar". The temporal units are self-explanatory, but there is more variety in how an amount of euros is displayed, e.g.: "2000 euro", "€ 87,66", "€ 87", "87,–", "87,=", "€ 2.662,63". A complication is that our regular expression requires the unit corresponding to the matched digit to come *after* the digit. This is only explicit in the case of "x *euro*." In the other cases, we need to infer that a number refers to an amount of euros. Because euro signs appear in front of the number, they are not matched as a unit (a positive lookback does not suffice in this case, because the euro sign is already consumed by the connector regex). In cases where the matching digit is succeeded e.g. by ",66" or ",–", then these suffixes

indicate that we are dealing with money and are matched as the unit. We apply similar reasoning for higher fines such as "€ 2.662,63", but in this case ".662,63" is matched as the suffix. These cases can be easily handled in the downstream logic by checking if the unit starts with a comma or period and if so, reconstruct the total fine by appending the suffix (with some additional preprocessing) to the matched digit.

This approach catches all scenarios except "€ 20", in which case "0" is matched as the unit. If the unit consists only of digits like in this case and occurs after an indication for a fine, then we parse it as a fine, unless the digit occurs in a context that we filter out as an exception, such as when it indicates a numbered fact ("feit") or a date. These exceptions include dates (e.g. "27-01-2021") and numeric identifiers ("parketnummer 23/003276-17" or "artikel 6:6:25 van Wetboek van Strafvordering"). These are handled downstream by discarding matches with a unit starting with "-" or "/" or ":". Note that we could have opted for extending the negative lookahead (currently only for "feit") behind the first digit. This would however only move the problem because e.g. in "27-01-2021" this would match "2" as the number and "7" as the unit. Similarly, in "2-01-2021" this would skip the first "2", but then match the next "2" and the "1" as the unit. The benefit of the chosen approach is that we can now consistently recognize if we are dealing with an identifier or date downstream, because the symbols such as ":" are included in the match. Another check is to control that no month names occur between matching digits, such as in "betaling van dit bedrag, vermeerderd met de wettelijke rente over dit bedrag vanaf 7 mei 2016 tot aan" (ECLI:NL:RBNHO:2020:11590).

Because we limited the scope to main punishments (with the exception of TBS), we do not want to match another type of payment, such as the measure to return unlawfully obtained advantages (goods or money). We perform additional checks for the occurrence of "wederrechtelijk verkregen voordeel" (unlawfully obtained advantages), "schadevergoeding" or "smartegeld" (compensation), as well as checking for "maatregel" (measure), using the "test" and "modifier" named capture groups.

When a fine is given, the sum of money is assigned to the person(s) making the claim ("vordering"). However, in Dutch law the sum is not directly paid to the claimant, but instead to the state which, in turn, pays out the sum to the claimant. This means that strictly speaking there are two money transactions for a single fine. This is also linguistically represented in case decisions: first we'll see that the fine is awarded to the claimant, but secondly we see an explicit statement that the defendant is obliged to pay that same sum to the state. We want to match the first case, but not the second in order to avoid doubling the amount of fines. Many of the duplicate instances have a phrasing that is not matched by the regular expression (e.g. "Legt verdachte de verplichting op ten behoeve van [naam slachtoffer] aan de Staat € 5.000,- (vijfduizend euro) te betalen"), but this does not always hold. We do another check on presence of "aan de Staat" to avoid matches such as: "bepaalt dat verdachte verplicht is ter zake van het bewezen verklaarde feit tot betaling aan de Staat der Nederlanden van een bedrag van € 436,27".

Sentences with components `?:(?P<nummer2>\d+(?!\s?\]))[^0-9\n\r;.]{0,30}?(?:\s(?P<eenheid2>jaar|jaren|maanden|maand|week|weken|dagen|dag|uur|uren)))?`

In particular for punishments such as prison sentences, the sentence is typically written out explicitly as such: "gevangenisstraf van 12 (twaalf) maanden en 6 (zes) maanden". We repeat a similar procedure as described above for the additional sentence component, the main difference being that we only accept temporal units and that the second component is an optional match.

Modifiers `?:(?P<modifier1>voorwaardelijk|proeftijd|niet|vervangend|indien|mindering|maatregel)[^\n\r;.]{0,100})?`

`?P<modifier2>voorwaardelijk|proeftijd|niet|vervangend|indien|hechtenis|wederrechtelijk))?`

Finally, we match several keywords, which we call "modifiers," that are important for the interpretation of the match. These modifiers are optionally matched either at the beginning or the end of the regular expression. It is possible that a particular punishment is mentioned precisely because it will *not* be imposed. The following examples are some negations from the data set that are detected such that the mentioned punishments are discarded: "verklaart de benadeelde partij niet ontvankelijk in de vordering ter hoogte van € 1.176,50 voor de kosten van de huishoudelijke hulp" (ECLI:NL:RBROT:2020:11499); "detentie groot 30 (dertig) dagen, niet ten uitvoer zal worden gelegd, tenzij de rechter later anders mocht gelasten wegens niet" (ECLI:NL:RBROT:2020:12683); "bij niet betaling te vervangen door 204 dagen gijzeling, met dien verstande dat toepassing van de gijzeling de betalingsverplichting niet opheft" (ECLI:NL:RBZWB:2020:6268).

A limitation of the regex is that these negations are only matched at the beginning or end of a match. We therefore do an additional check on the keywords "niet", "niet ten uit" and "niet tenuit" in the connector groups called "niettest1" and "niettest2," to catch edge cases where the negation occurs somewhere else. This is also necessary due to the connector group between "eenheid1" and "nummer2" being greedy: this often consumes "niet" because the ending modifier group is optional. Making this connector group lazy introduces a nasty side-effect that prevents matching the optional later parts of the regex (such as "en 6 (zes) maanden"): it will first look for matches without expanding; which succeeds because the rest of the string can be matched by the "niettest2" group. There is no "incentive" to backtrack and look for "nummer2" instead. Due to this extra check, the following case is correctly recognized as *not* being executed: "bepaalt dat van deze gevangenisstraf een gedeelte, groot 10 (tien) weken, niet ten uitvoer zal worden gelegd, tenzij de rechter later anders mocht gelasten;" (ECLI:NL:RBROT:2020:13002).

The keywords "voorwaardelijk" (conditional) and "proeftijd" (probation) indicate that (a part of) a sentence is conditional. This is a more tricky case, because in some cases we arguably want to include a conditional punishment and in others not. In the following example, a conditional prison sentence is part of the punishment: "gevangenisstraf van 2 weken voorwaardelijk met een proeftijd (...)" (ECLI:NL:RBZWB:2020:6449). However, in the following cases we do *not* want to add the conditional part of the sentence: "gevangenis-

straf van 12 (twaalf) maanden, waarvan 6 (zes) maanden voorwaardelijk met een proeftijd" (ECLI:NL:RBZWB:2020:6775); "gevangenisstraf van 18 (achttien) maanden, waarvan 3 (drie) maanden voorwaardelijk met een proeftijd (...)" (Case: ECLI:NL:RBZWB:2020:6166). This is a very common phrasing. If the optional second part of a sentence has the modifier "voorwaardelijk" or "proeftijd" we therefore exclude this part from the total sentence sum.

In addition to specifying conditional punishments or a conditional part to an otherwise non-conditional punishment, it is also very common to specify a replacement punishment in case some other requirement is not fulfilled. The keywords "indien" (if) and "vervangend" (replacing) and "subsidiair" (subsidiary) indicate that a sentence is a subsidiary punishment in case the main punishment is not executed properly. We are interested in the original sentence and skip these matches. The matching punishments in the following example is discarded: "indien verdachte de taakstraf niet naar behoren verricht, vervangende hechtenis zal worden toegepast van 30 dagen" (ECLI:NL:RBZWB:2020:6449; N.B. in this example "hechtenis" is correctly matched as the punishment, instead of "taakstraf"). The subsidiary punishment is typically a form of custody ("hechtenis") in case a community service or a fine has not been paid. In the following case it is not sufficient to check for "vervangend" in the ending modifier: "taakstraf van 80 (tachtig) uren, met bevel dat indien deze straf niet naar behoren wordt verricht vervangende hechtenis zal worden toegepast voor de duur van 40 (veertig) dagen". In this case we correctly recognize a community service of 80 hours in "taakstraf van 80 (tachtig) uren, met bevel dat indien deze straf niet naar behoren wordt verricht vervangend", but then incorrectly start a new match at "hechtenis" so that the subsidiary custody is counted as a punishment of its own: "hechtenis zal worden toegepast voor de duur van 40 (veertig) dagen". This case is handled by also matching "hechtenis" as an ending modifier if it occurs in reasonably close vicinity to a previously mentioned punishment, which is appropriate because it is so commonly a replacement punishment. This will then capture "taakstraf van 80 (tachtig) uren, met bevel dat indien deze straf niet naar behoren wordt verricht vervangende hechtenis" and avoids a further match starting with the subsidiary custody. In some instances a subsidiary punishment is mentioned in the past tense, as in: "hechtenis heeft doorgebracht naar rato van 50 euro per dag". Detection of "[heeft|is] doorgebracht" will lead to rejection of the match. This works well enough because legal experts tend to use the exact same phrasings over and over, but conceptually speaking an improvement would be to apply a parser to detect the past tense in matches.

It is also very common that a suspect has already been detained while awaiting the case to appear in court. The judge's verdict will, for legal reasons, explicitly mention that this custody will be detracted from the total punishment, if applicable. In order to avoid *adding* a punishment that is in fact *detracted* we check for the phrase "in mindering", either by matching it as the first modifier or in the "test1" capture group. We additionally test for the presence of "heeft doorgebracht" or "is doorgebracht" in the "test1" group. This avoids incorrectly detecting a community service for example in the following sentence: "bepaalt dat de tijd die verdachte voor de tenuitvoerlegging van deze uitspraak in vooarrest heeft doorgebracht in mindering wordt gebracht bij de tenuitvoerlegging van de taakstraf naar rato van 2 uur per dag".

Finally, we have domain knowledge on the maximum height of sentences (30 years for prison sentence, 1 year for custody, and 10 days for community service). If we find a punishment that exceeds the legally allowed maximum, this is certainly due to a parsing error. We detect these cases we clip the height to the maximum value and raise a warning, so that these data points may later be filtered out if desired.

1.2 TBS

```
(?i)(?:(?P<verlenging>verlengt|verlenging).{0,50})?(?P<TBS>TBS|te
rbeschikkingstelling|ter␣beschikking␣(?:wordt␣|is␣)?(?:stelling|g
esteld))(?:(?!voorwaarde|verple).){0,100}(?P<type>voorwaarden|ver
pleging|verpleegd)?
```

A notable difference with the main punishments is that phrases imposing TBS do not explicitly specify a duration, because TBS is imposed for 2 years and then re-evaluated and possibly extended. There are two types of TBS: either TBS is imposed with mandatory nursing ("dwangverpleging" of "verpleging van overheidswege") or with a set of conditions ("met voorwaarden") such as a stay in a forensic psychiatric hospital or taking certain medications. The regex therefore requires either an indication for mandatory nursing or an indication of conditions, next to the indication for TBS. However, there are also often cases where it is decided to prolong a TBS measure that has previously been imposed. In these cases there is often an indication of the duration, but this is typically until the next review moment, which is a standardized moment and hence not interesting to parse.

The phrase "ter beschikking stellen" is heavily used in legal jargon in different contexts as well, so in order to avoid false positives we do require either an indication of prolongation of a previous TBS measure, or an indication of the type. We may either enforce this in the regular expression itself or handle this downstream in the rule-based classifier. If enforcing this in the regular expression itself, the expression would have the logical form:

$(verlenging \wedge TBS) \vee (TBS \wedge type)$

This duplicates the TBS expression and on average will require the regex engine to take significantly more steps for each match. We instead opted for an expression with an optional "verlenging" group before the TBS expression and an optional "type" group after it, i.e. we capture all possible relevant information and discard matches when they do not meet our requirements. In this case, the rule-based classifier discards a match when neither of the optional groups is matched.

The following examples are all detected: "gelast de terbeschikkingstelling van verdachte, met verpleging van overheidswege;" (ECLI:NL:RBZWB:2020:6268); "gelast dat de verdachte, voor de feiten 2, 3 en 4, ter beschikking wordt gesteld en stelt daarbij de volgende, het gedrag van de ter beschikking gestelde betreffende, voorwaarden" (ECLI:NL:RBLIM:2020:9778); "De rechtbank verlengt de termijn van de terbeschikkingstelling van veroordeelde met één jaar." (ECLI:NL:RBNNE:2020:4558); "verlengt de termijn gedurende welke [verdachte] ter beschikking is gesteld met verpleging van overheidswege met één jaar" (ECLI:NL:RBLIM:2020:10468).

1.3 Acquittal

```
(?i)(?P<vrijspraak>vrijgesproken|vrijspraak|spreekt[^.\r\n;]*\svr
ij|wijst[^\r\n;]{0,100}\saf)(?:(?!meer␣of␣anders).){0,50}(?P<nebi
sinidem>meer␣of␣anders␣(?:ten␣laste␣is|is␣ten␣laste)␣gelegd)?
```

This regular expression is more straightforward. Consider the following representative examples: "spreekt de verdachte vrij" (ECLI:NL:RBLIM:2020:9690); "wijst de vordering van de benadeelde partij voor het overige af" (ECLI:NL:RBLIM:2020:9690); "wijst de vordering tot immateriële schade voor het overige af" (ECLI:NL:RBGEL:2020:6988).

The phrase "wijst voor het overige af" (rejection of remaining claims) is a construct that can be used to summarize acquittal on several claims. In many cases we find a similar phrase which, however, has a different special significance. For example, in case ECLI:NL:RBZWB:2020:6800 we find:

"- verklaart het ten laste gelegde bewezen, zodanig als hierboven onder 4.4 is omschreven; - spreekt verdachte vrij van wat meer of anders is ten laste gelegd;" (acquittal of all points except the proven facts mentioned in section 4.4).

The facts in section 4.4 are, however, the only facts mentioned in the case. This phrasing rather indicates that there will be no repeated prosecution on closely related facts. This is called the *ne bis in idem* ("not twice in the same") principle in civil law; its equivalent being the "double jeopardy" clause in common law. This principle states that one cannot be persecuted twice for the same facts, although there may be dispute about which courses of actions constitute new facts and which do not. This principle applies when a fact has already been judged by a foreign judge or a Dutch judge, or when a settlement is agreed upon and the suspect has paid. A fact is considered to be judged when at least one material question (as opposed to the formal questions, such as the determination whether the fact is punishable) has been answered by the judge.

In any case, the relevance for the labelling of case outcomes is that even though this phrase linguistically resembles acquittal, it is equally applicable in situations where a punishment is imposed as in those where the suspect is acquitted of all charges. This means that we should recognize this situation and not label it as "acquittal". This is done with the following expression:

```
(?:(?!meer␣of␣anders).){0,50}(?P<nebisinidem>meer␣of␣anders␣(?:te
n␣laste␣is|is␣ten␣laste)␣gelegd)?
```

Because the "ne bis in idem" clause is optional, it is necessary to temper the scope of the `.` token with a negative lookahead. This avoid that the construct `.0,50` expands over the optional group and never matches it, even if it is present. This construct is also used in the regular expression for TBS.

1.4 Test cases

To provide insight into the behavior of the regular expressions and the rule-based classifier defined on the matches, we provide test cases that represent several scenarios. A full suite of tests can be found in the code repository.

Simple cases The following cases represent typical situations. The duration of a punishment is practically always mentioned with a digit, alongside the

number fully written out. The duration is almost without exception mentioned *after* a mention of the type of punishment. Punishments are converted to days or euros, except for "TBS" and "vrijspraak" which are binary.

```
TEST CASE: een gevangenisstraf van 5 (vijf) jaren
MATCH GEVANGENISSTRAF: gevangenisstraf van 5 (vijf) jaren
OUT: {'TBS': 0, 'gevangenisstraf':  1825, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

```
TEST CASE: veroordeelt de verdachte tot hechtenis voor de duur van
3 (drie) maanden;
MATCH HECHTENIS: hechtenis voor de duur van 3 (drie) maanden
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  91, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

Community service is often expressed in terms of hours. In this case we round up to an integer amount of days.

```
TEST CASE: Persoon wordt veroordeeld tot taakstraf van 40 (veertig)
uur
MATCH TAAKSTRAF: taakstraf van 40 (veertig) uur
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
2, 'geldboete':  0, 'vrijspraak':  0}
```

Compounded durations In the case of prison sentences, the duration is often compounded. A duration of 2.5 year is written as "2 (two) years and 6 (six) months", so requires two matches of the time unit and conversion of both units into days in order to sum them.

```
TEST CASE: veroordeelt de verdachte tot een gevangenisstraf voor de
duur van 2 (twee) jaar en 6 maanden;
MATCH GEVANGENISSTRAF: gevangenisstraf voor de duur van 2 (twee) jaar
en 6 maanden
OUT: {'TBS': 0, 'gevangenisstraf':  912, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

Conditions, probation, subsidiary punishments However, in several instances we find two units of time that we do not want to sum up together. A very similar situation is where a conditional part of an unconditional punishment is specified. We should not add the conditional part to the sentence length.

```
TEST CASE: veroordeelt verdachte tot een gevangenisstraf van 12 (twaalf)
maanden, waarvan 6 (zes) maanden voorwaardelijk met een proeftijd van
twee jaar;
MATCH GEVANGENISSTRAF: gevangenisstraf van 12 (twaalf) maanden, waarvan
6 (zes) maanden voorwaardelijk met een proeftijd
Conditional punishment detected.
WARNING: Second part of sentence is conditional.  This part is excluded.
OUT: {'TBS': 0, 'gevangenisstraf':  365, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

We often have a situation where the probation period of a sentence is explicitly mentioned.

```
TEST CASE: voorwaardelijke gevangenisstraf van 5 jaar, en verbindt
hieraan een proeftijd, die wordt gesteld op 2 jaar;
MATCH GEVANGENISSTRAF: voorwaardelijke gevangenisstraf van 5 jaar,
en verbindt hieraan een proeftijd
Conditional punishment detected.
OUT: {'TBS': 0, 'gevangenisstraf':  1825, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  }
```

For fines and community service we often find a replacement or subsidiary punishment in case the first one is not fulfilled. When this is not recognized this can lead to very wrong results, such as community service in the order of 30 or 50 days, whereas the legal maximum is 10 days.

```
TEST CASE: taakstraf bestaande uit het verrichten van onbetaalde arbeid
voor de duur van 60 (zestig) uren, subsidiair 30 dagen hechtenis
MATCH TAAKSTRAF: taakstraf bestaande uit het verrichten van onbetaalde
arbeid voor de duur van 60 (zestig) uren, subsidiair 30 dagen hechtenis
Subsidiary punishment detected.  This part is excluded.
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
3, 'geldboete':  0, 'vrijspraak':  0}
```

```
TEST CASE: wanneer taakstraf niet naar behoren heeft verricht, wordt
vervangende hechtenis toegepast van 50 (vijftig) dagen
MATCH TAAKSTRAF: taakstraf niet naar behoren heeft verricht, wordt
vervangende hechtenis toegepast van 50 (vijftig) dagen
Subsidiary punishment detected.  Skipped.
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

Reduction Often a suspect will have spent time in custody 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.

```
TEST CASE: bepaalt dat de tijd die verdachte voor de tenuitvoerlegging
van deze uitspraak in voorarrest heeft doorgebracht in mindering wordt
gebracht bij de tenuitvoerlegging van de taakstraf naar rato van 2
uur per dag;
MATCH TAAKSTRAF: mindering wordt gebracht bij de tenuitvoerlegging
van de taakstraf naar rato van 2 uur per dag
'in mindering' detected.  Skipped.
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

In some cases the judge will mention time already spent in custody in past tense without the key words "in mindering". We also filter these out.

```
TEST CASE: hechtenis heeft doorgebracht naar rato van 50 euro per dag
MATCH HECHTENIS: hechtenis heeft doorgebracht naar rato van 50 euro
per dag
Previous punishment detected.  Skipped.
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
```

```
0, 'geldboete':  0, 'vrijspraak':  0}
```

Negations It occurs regularly that a (part of a) sentence is mentioned precisely because it is *not* executed. These negations should be detected.

```
TEST CASE: gevangenisstraf een gedeelte van 120 (honderdtwintig) dagen
niet ten uitvoer
MATCH GEVANGENISSTRAF: gevangenisstraf een gedeelte van 120 (honderdtwintig)
dagen niet ten uit
Negation ('niet ten uitvoer') detected.  Skipped.
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

```
TEST CASE: van deze gevangenisstraf zal een gedeelte, groot 3 (drie)
maanden, van deze gevangenisstraf niet tenuitvoergelegd worden, tenzij
later anders wordt gelast.  Stelt daarbij een proeftijd van 2 (twee)
jaren vast.
MATCH GEVANGENISSTRAF: gevangenisstraf zal een gedeelte, groot 3 (drie)
maanden, van deze gevangenisstraf niet
Negation detected at beginning or end of match.  Skipped.
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

Fines Fines may be written in a variety of ways, e.g. as '€5.000,-', '€60', '500,= euros' and so on. We discard cents.

```
TEST CASE: Wijst de vordering van de benadeelde partij [naam slachtoffer]
toe tot een bedrag van € 5.000,- (vijfduizend euro) aan vergoeding
van immateriële schade
MATCH VORDERING: vordering van de benadeelde partij [naam slachtoffer]
toe tot een bedrag van € 5.000,- (vijfduizend e
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  5000, 'vrijspraak':  0}
```

Often, the components of the total fine are also mentioned. We want to avoid matching and summing the subcomponents as well, otherwise we will double the height of the fine.

```
TEST CASE: Wijst de vordering van de benadeelde partij [slachtoffer
1] toe tot een bedrag van € 1.314,28 (duizend driehonderdveertien euro
en achtentwintig cent), bestaande uit € 314,28 (driehonderdveertien
euro en achtentwintig cent) aan vergoeding van materiële schade en
€ 1.000,00 (duizend euro) aan vergoeding van immateriële schade, te
vermeerderen met de wettelijke rente daarover vanaf het moment van
het ontstaan van de schade op 27 juni 2020 tot aan de dag van de algehele
voldoening.
MATCH VORDERING: vordering van de benadeelde partij [slachtoffer 1]
toe tot een bedrag van € 1.314,28 (duizend drieh
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  1314, 'vrijspraak':  0}
```

When a fine is imposed on the suspect, the suspect has to pay the fine to

the state on behalf of the person making the claim. Often this obligation to pay the state with the same amount is mentioned separately, which again is a risk for doubling the height of the fine. This is an important reason for not just using a more simple regular expression for matching all amounts of money, but only matching amounts of euros in certain conditions. The imposed fine is typically preceded by the term "vordering" (claim), whereas the payment to the state that we do not want to match is preceded by another term such as "verplichting" (obligation).

```
TEST CASE: Wijst de vordering van de benadeelde partij [naam slachtoffer]
toe tot een bedrag van € 5.000,- (vijfduizend euro) aan vergoeding
van immateriële schade.  Legt verdachte de verplichting op ten behoeve
van [naam slachtoffer] aan de Staat € 5.000,- (vijfduizend euro) te
betalen.
MATCH VORDERING: vordering van de benadeelde partij [naam slachtoffer]
toe tot een bedrag van € 5.000,- (vijfduizend e
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  5000, 'vrijspraak':  0}
```

Sometimes it is explicitly mentioned that we are dealing with a measure instead of a punishment. If the "modifier1" group matches "maatregel" we discard the match. If "maatregel" is absent but we see a phrase indicating payment to the State ("aan de Staat"), then we are also almost certain the mentioned sum is duplicated and discard the match.

```
TEST CASE: legt de maatregel op dat verdachte verplicht is ter zake
van het bewezen verklaarde feit tot betaling aan de Staat der Nederlanden
van een bedrag van € 436,27, te vermeerder
MATCH BETALING: maatregel op dat verdachte verplicht is ter zake van
het bewezen verklaarde feit tot betaling aan de Staat der Nederlanden
van een bedrag van € 436,27, te vermeerder
Measure ('maatregel') detected.  Skipped
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

```
TEST CASE: dat verdachte verplicht is ter zake van het bewezen verklaarde
feit tot betaling aan de Staat der Nederlanden van een bedrag van €
436,27, te vermeerderderen
MATCH BETALING: betaling aan de Staat der Nederlanden van een bedrag
van € 436,27, te vermeerder
Payment to state detected.  Probably duplicated sum.  Skipped.
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

TBS TBS ("terbeschikkingstelling") is imposed with conditions ("voorwaarden"), mandatory nursing (e.g. "verpleging van overheidswege"), or prolongation ("verlenging").

```
TEST CASE: gelast dat de verdachte, voor de feiten 2, 3 en 4, ter beschikking
wordt gesteld en stelt daarbij de volgende, het gedrag van de ter beschikking
gestelde betreffende, voorwaarden (ECLI:NL:RBLIM:2020:9778)
MATCH TBS: ter beschikking wordt gesteld en stelt daarbij de volgende,
```

```
het gedrag van de ter beschikking gestelde betreffende, voorwaarden
OUT: {'TBS': 1, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

```
TEST CASE: verlengt de termijn gedurende welke [verdachte] ter beschikking
is gesteld met verpleging van overheidswege met één jaar" (ECLI:NL:RBLIM:2020:10468)
MATCH TBS: verlengt de termijn gedurende welke [verdachte] ter beschikking
is gesteld met verpleging
OUT: {'TBS': 1, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

"Ter beschikking stellen" is used in other legal contexts as well, so we do not match this term by itself if it occurs without conditions, mandatory nursing, or prolongation.

```
TEST CASE: ter beschikking stelling van de goederen aan benadeelde
partij
MATCH TBS: ter beschikking stelling van de goederen aan benadeelde
partij
WARNING: neither 'verlenging' nor 'type' of TBS detected.  Skipped.
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

```
TEST CASE: TBS kliniek De Kijvelanden, wederrechtelijk van de vrijheid
heeft beroofd en/of beroofd gehouden, immer
MATCH TBS: TBS kliniek De Kijvelanden, wederrechtelijk van de vrijheid
heeft beroofd en/of beroofd gehouden, immer
WARNING: neither 'verlenging' nor 'type' of TBS detected.  Skipped.
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

Acquittal Acquittal on some fact is relatively easy to match.

```
TEST CASE: spreekt de verdachte daarvan vrij
MATCH VRIJSPRAAK: spreekt de verdachte daarvan vrij
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  1}
```

```
TEST CASE: wijst de vordering van de benadeelde partij voor het overige
af
MATCH VRIJSPRAAK: wijst de vordering van de benadeelde partij voor
het overige af
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  1}
```

In some cases we find an indication of acquittal that is a formal statement that excludes further prosecution on the same facts. The following case finds the suspect guilty on all charges but contains a formal phrase that indicates this "ne bis in idem" principle (cf. "no double jeopardy").

```
TEST CASE: - verklaart het ten laste gelegde bewezen, zodanig als hierboven
onder 4.4 is omschreven; - spreekt verdachte vrij van wat meer of anders
is ten laste gelegd;
```

```
MATCH VRIJSPRAAK: spreekt verdachte vrij van wat meer of anders is
ten laste gelegd
'ne bis in idem' detected.  Skipped.
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

No measures In particular amounts of money are easily confused with measures as opposed to punishments. Because we have limited our scope to main punishments, we filter out measures where possible for the sake of consistency.

```
TEST CASE: legt [verdachte] de verplichting op tot betaling aan de
staat ter ontneming van het wederrechtelijk verkregen voordeel van
€ 331.083,14 (zegge:  driehonderdeenendertigduizend drieëntachtig euro
en veertien eurocent)
MATCH BETALING: betaling aan de staat ter ontneming van het wederrechtelijk
verkregen voordeel van € 331.083,14 (zegge:  drieho
Measure to return unlawfully obtained advantages detected.  Skipped.
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

```
TEST CASE: veroordeelt verdachte in verband met het feit onder nummer
1 en 2 tot betaling van schadevergoeding aan de benadeelde partij [getuige
1] van 37,48 aan materiële schade en 1.500,- aan smartengeld, vermeerderd
met de wettelijke rente vanaf 22 november 2019 tot aan de dag dat het
hele bedrag is betaald
MATCH BETALING: betaling van schadevergoeding aan de benadeelde partij
[getuige 1] van 37,48 aan materiële Measure for compensation detected.
Skipped.
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

Edge cases We avoid matching dates and case identifiers as fines.

```
TEST CASE: vordering van de officier van justitie tot tenuitvoerlegging
in de zaak met parketnummer 23/003276-17
MATCH VORDERING: vordering van de officier van justitie tot tenuitvoerlegging
in de zaak met parketnummer 23/003276-
Identifier detected, e.g.  a date, case number, law reference.  Skipped.
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

```
TEST CASE: veroordeelt verdachte tot betaling van het toegewezen bedrag
voor de schade ontstaan op 4 juli 2020
MATCH BETALING: betaling van het toegewezen bedrag voor de schade ontstaan
op 4 juli 2020
WARNING: Date detected with month juli.  Skipped
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

We often find digits that enumerate certain charges or facts and that should not be matched as the height of a punishment. These are skipped with a negative lookbehind.

```
TEST CASE: betaling aan de benadeelde partij [Slachtoffer 3] (feit
9) van € 5.226,53,
MATCH BETALING: betaling aan de benadeelde partij [Slachtoffer 3] (feit
9) van € 5.226,53,
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  5226, 'vrijspraak':  0}
```

1.5 Failure analysis

Regular expressions are surprisingly effective for extracting the sentences from case decisions because juridical language use is relatively structured. There are however some exceptions that the regular expressions do not capture. The good f1-score of 0.94 shows that the following situations are indeed exceptions. It is possible to extend the used regular expressions and the rule-based classifier, but within the scope of this project a labelling score of 0.94 is more than sufficient.

The used regular expression assumes that the name of the type of punishment precedes the duration of the punishment. Cases where this order is reversed are not matched or wrongly matched in some edge cases:

```
TEST CASE: Veroordeelt verdachte tot honderdtachtig (180) dagen jeugddetentie.
OUT: 'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0
NO MATCHES FOUND
```

```
 TEST CASE: Gelast de tenuitvoerlegging van de werkstraf, voor zover
voorwaardelijk opgelegd bij vonnis van de kinderrechter van Rechtbank
Noord-Nederland, locatie Leeuwarden van 6 oktober 2020, te weten:  50
uren werkstraf subsidiair 25 dagen vervangende jeugddetentie
MATCH WERKSTRAF: werkstraf subsidiair 25 dagen vervangende
OUT: 'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
2, 'geldboete':  0, 'vrijspraak':  0
```

We also assume the length of the punishment is indicated with a digit, but because the anonymization scheme of rechtspraak.nl uses digits within brackets (e.g. '[feit 1]' or '[persoon 1]') we avoid matching digits followed by a square closing bracket. In very rare instances, we may see stylistic inconsistencies that prevent matching. I have found at least one case where use of squares instead of regular brackets for the sentence length avoided a match. This match is rejected because the "nummer1" regex group matches "2" and the "eenheid1" regex group matched "4", which normally would indicate we are dealing with an amount of euros. The rule-based classifier then complains that we find a prison sentence with a fine, which is inconsistent.

```
TEST CASE: gevangenisstraf voor de duur van vierentwintig [24] maanden
MATCH GEVANGENISSTRAF: gevangenisstraf voor de duur van vierentwintig
[24] maanden
Amount of euros found, but punishment is not a fine
OUT: {'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0}
```

In some cases no digit is mentioned at all. Recognizing numbers written out

in natural language would require a parser; regular expressions are not a good solution for this. Luckily, this situation occurs rarely.

```
TEST CASE: veroordeelt verdachte tot een taakstraf van honderdtwintig
uren;
OUT: 'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  0, 'vrijspraak':  0
NO MATCHES FOUND
```

Typically, we correctly match the total amount of money without adding up its components, but if the components are mentioned first after the indication of punishment we incorrectly match the subcomponent as the total:

```
TEST CASE: vordering van de benadeelde partij [persoon] toe tot een
bedrag van € 87,- (zevenentachtig euro) aan vergoeding van materiële
schade en € 1.000 aan immateriele.
MATCH VORDERING: vordering van de benadeelde partij [persoon] toe tot
een bedrag van € 87,- (zevenentachti
OUT: 'TBS': 0, 'gevangenisstraf':  0, 'hechtenis':  0, 'taakstraf':
0, 'geldboete':  87, 'vrijspraak':  0
```