

# Automatic Classification of Sentences in Dutch Laws

Emile de Maat and Radboud Winkels  
*University of Amsterdam*  
*Leibniz Center for Law*  
*{demaat, winkels}@uva.nl*

## **Abstract.**

The work described here builds on [1], where we presented a categorisation of norms or provisions in legislation. We claimed that the categories are characterized by the use of typical sentence structures and that this would enable automatic detection and classification. In this paper we present the results of experiments in such automatic classification of provisions. We have defined fourteen different categories of provisions, and compiled a list of 81 sentence structures for those categories from twenty Dutch laws. Based on these structures, a parser was used to classify the sentences in fifteen different Dutch laws, classifying 94% of 530 sentences correctly. It compares well with other, statistical approaches. An important improvement of our classifier will be the distinction of principal and auxiliary sentences.

**Keywords:** automatic classification; natural language processing.

## **1. Introduction**

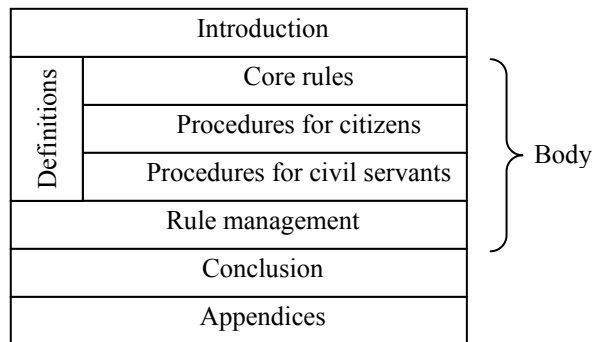
These last years we have been researching the possibility to support legal knowledge engineers in making rich, complex and *isomorphic* models of sources of law with the use of natural language processing (NLP) techniques. In the (E-)POWER project<sup>1</sup> we experimented with parsers to suggest UML/OCL representations for some fragments of legislation (see also [2]). In [3] we described the use of similar technology to detect and represent references in and between sources of law. The work described here builds on [1], where we presented a categorisation of norms or provisions in a major source of law: legislation. We claimed that the categories are characterized by the use of typical sentence structures and that this would enable automatic detection. In this paper we present the results of experiments in such automatic classification of provisions, but first we start with a short summary of the types of provisions we distinguish. Then we will describe the experimental setup, present results and end with some conclusions and discussion.

---

<sup>1</sup> <http://www.lri.jur.uva.nl/~epower/>

## 2. Approach

The figure below presents an overview of the types of provisions or sentences we distinguish in legislation [1]. The ‘Introduction’, ‘Conclusion’ and ‘Appendices’ are relatively unimportant for most uses of legislation. The other types together form what we have called the ‘body’ of the law. Important for most uses are the so called ‘core rules’ that regulate what the legislator intended to regulate in society, e.g. the rights and obligations of unemployed people in a social welfare law. In order to obtain and handout an unemployment benefit in this example, the law also contains procedures for citizens and civil servants. The ‘core rules’ and the procedures make use of vocabulary that is partly specifically defined in the law, the ‘definitions’. Finally, there typically are auxiliary provisions to fit all others in the legal systems as a whole, e.g. an enactment clause.



We are interested in the body of the document, and would like to be able to classify the sentences that appear in the document according to their meaning. For several types of sentences, we can distinguish certain signal words, certain patterns, which tell us what kind of sentence we are dealing with. However, a problem occurs when we encounter obligations and prohibitions. Though these are sometimes expressed with words like “should” or “must”, most of the time a ‘statement of fact’ is used<sup>2</sup>. This means that the text does not state what must happen, but instead simply states that it happens. For example:

**Funeral Act, article 46, sub 1**  
No bodies are interred on a closed cemetery.

Obviously, such statements have few words in common with similar statements from a different domain. Thus, there is no pattern to be found either. Because of this, we hope to identify this important group of statements “by default”: if we can identify patterns for everything else, we may assume that anything not classified by these patterns is one of these statements of fact.

As for the other sentence types, it was found that the general type of the sentence could usually be derived from the verb phrase of the sentence. In earlier experiments,

<sup>2</sup> The official guidelines [5] strongly advise legislative drafters to use this form, and to avoid words like “must”.

we used more elaborate patterns, consisting of the verb phrase with some other words appended, but it would always turn out that these patterns were too restrictive. For many applications, when dealing with verbs, a stemming algorithm would be employed to deal with different inflections for tense, person and number. However, in legislation, the tense does not vary a lot (with present simple being used most of the time) and the rules always use third person<sup>3</sup>. As such, we deemed it unnecessary to employ something more complex than a simple pattern recogniser.

### 3. Patterns

Earlier research [4] into the types of sentences that occurs in laws formed the basis for the patterns. However, this earlier research was largely based on a single (though extensive) law, the Income Tax Act 2001. In addition, as it was part of the (E-)POWER project, it focused on the core rules and procedures expressed in the law, and did not pay attention to the rule management part. Hence, the patterns needed to be extended. About 20 other laws were studied to extend the set of patterns found in earlier research.

#### 3.1. Definitions and type extensions

In a definition, a description is given of the terms that are used in the legal source. The construct that is used for a definition in Dutch legal texts is: *by x is understood y*, which gives us a clear pattern to identify definitions by. Type extensions are added definitions, which expand or limit an earlier definition, using the same verb phrase, but with the inclusion of the word *also* or *not*.

In the earlier research on the Income Tax Act 2001, the patterns used were: *x is y*, or: *x are y* [4]. So far, however, our research indicates that the Income Tax Act was somewhat unique in its use of those patterns, and we have not included them here (meaning that this classifier would not work well on the Tax Income Act). Should it turn out that these patterns are more widely spread, a more advanced classifier would be necessary to distinguish between definitions using this pattern and statement of fact sentences that merely use *is* or *are* as an auxiliary verb.

#### 3.2. Deeming provisions

Deeming statements are sentences in which a given situation is said to be considered as if it were another situation. Thus, if situation A is deemed equal to situation B, then all rules that apply to situation B apply to situation A as well. In this way, definitions can be extended to cover certain special, exceptional situations. Deeming provisions can be recognized by the pattern: *is deemed to*.

#### 3.3. Norms

In preparation of this project to classify all sentences in a legal text, research was conducted to determine how to distinguish between norm sentences denoting a right

---

<sup>3</sup> The introduction and the conclusion which contain the directions from the King or Queen will employ first person plural (majestic plural).

and those sentences denoting a duty [6]. This work yielded a large amount of patterns that were incorporated in the classifier for this project. The main conclusion was that almost 80% of the rights could be identified by the verb: *may* (Dutch: *kunnen*) or the phrase: *is qualified* (Dutch: *is bevoegd*). A host of smaller patterns accounted for the remaining rights.

Sentences denoting duties usually did not follow a pattern; 80% of these sentences was a statement of fact (as described in section 2).

#### 3.4. Application provisions

Application provisions are sentences that specify cases in which some other legislation (usually an article or subsection of an article) does (not) apply. In this way, additional conditions are added to existing norms. In case of an application provision that states that the other legislation does not apply, the application provision does, in fact, state an exception to that rule. An application provision that states that another piece of legislation does apply often seems to be in place to take away any doubts as to whether it ought to apply or not.

The patterns used by these sentences are: *does apply* and: *does not apply*.

#### 3.5. Penalization

Phrases may also specify some penalty that will be incurred if a norm is violated. In Dutch law, this is usually done through sentences like: *will be punished with*. In general, these phrases will be followed by a provision that denotes whether the punishable fact is a *crime* or a *misdemeanour*.

#### 3.6. Value assignments and changes

Value assignments are used to give a value to a certain term in the text. These values can later be changed. These sentences express the formula used to calculate some value used in other sentences.

<b>Income Tax Act 2001, article 3.3, sub 1</b>
--

<i>Taxable wages are wages reduced with the employee's discount.</i>
--

These sentences use a range of mathematical operations (*to reduce, to increase*) and comparisons (*at most*) which in combination with the verb *to be* or *to amount to* can be used to detect them.

#### 3.7. Lifecycle

These sentences deal with the maintenance of the legal texts, keeping them up to date. Most of them deal with modifying existing legislation, by adding new text, modifying text or deleting/repealing text. In addition to the sentences that express such changes, there are sentences that determine the enactment date of a legal source. Most laws include one such sentence, in which they determine their own enactment date. Another type of lifecycle sentence is the citation title designation, in which a (shortened) official title for the source is set. This usually appears at the end of the law text.

#### 4. Experimental Set-up

We built a classifier (in Java) that takes well structured legal sources as input and tries to classify their sentences according to their type based on typical patterns associated with these types. The types and their patterns were described in the previous section. In total, we used 81 patterns from about twenty Dutch laws, consisting of verb phrases, often with some keywords added. Most patterns consisted of one to three words.

We tested the classifier on fifteen different Dutch laws of various types. Four of these fifteen were completely new laws; the others changed already existing laws (as is the more common situation). With the exception of a single Royal Decree, these were all bills, pending at Parliament. (In this they differed from the set used to derive the patterns, which were all acts that had already been passed.) The length of the laws varied from very short (7 sentences) to quite long (166 sentences on 23 pages A4); most were quite recent (patterns in the past have been different). All laws are listed in Table 1 below.

For this experiment, an assumption was made with regard to sentences with an embedded list, such as:

##### **Tobacco Act, article 1**

In this law, and in the stipulations based on it, is understood by:

- a. tobacco products: ... ;
- b. Our Minister: ... ;
- c. appendix: ... ;
- ...

Here, we assumed that classification can be based on the first part of the sentence, and that the list items are not needed for the classification. As input for our classifier, we used documents tagged in MetaLex<sup>4</sup>, in which both sentences and lists were marked.

To check whether clauses were classified correctly, all sentences and lists in all laws were also classified manually.

#### 5. Results

Table 1 gives the total number of sentences and lists in the sources we used and the number of these that were classified correctly. The last column indicates whether the

```
<sentence>The first member will become:  
<quote>  
  <subpart>  
    <index>1.</index>  
    <sentence>Rules concerning affairs of the Kingdom... </sentence>  
  </subpart>  
</quote>  
</sentence>
```

Figure 1: Simplified MetaLex example with quoted element.

law was a completely new one or a law that changed an already existing law. In these last types of law, the elements that are changed, repealed or inserted are marked as so called ‘quoted’ elements within MetaLex. The classifier also classifies the sentences and lists within these quoted elements. In the example in Figure 1 a simplified MetaLex structure is given of the use of such a quoted element. The classifier will both try to classify the sentence stating the change and the sentence that will become part of the altered law (“Rules concerning...”).

<i>Source</i>	<i>Sentence</i>			<i>List</i>			<i>Type</i>
	Total	Correct	%	Total	Correct	%	
Royal Decree Stb. 1945, F 214 (as modified per 01/01/2002)	25	24	96%	4	3	75%	New
Bill 20 585 nr. 2	31	31	100%	4	3	75%	New
Bill 22 139 nr. 2	22	22	100%	2	2	100%	New
Bill 27 570 nr. 4	21	20	95%	0			Change
Bill 27 611 nr. 2	11	11	100%	1	0	0%	Change
Bill 30 411 nr. 2	141	133	94%	25	24	96%	New
Bill 30 435 nr. 2	40	39	98%	4	4	100%	Change
Bill 30 583 nr. A	27	27	100%	0			Change
Bill 31 531 nr. 2	3	3	100%	0			Change
Bill 31 537 nr. 2	28	28	100%	3	3	100%	Change
Bill 31 540 nr. 2	7	7	100%	0			Change
Bill 31 541 nr. 2	8	8	100%	0			Change
Bill 31 713 nr. 2	7	6	86%	2	2	100%	Change
Bill 31 722 nr. 2	31	22	71%	6	6	100%	Change
Bill 31 726 nr. 2	74	67	91%	3	1	33%	Change
<b>Total</b>	<b>476</b>	<b>448</b>	<b>94%</b>	<b>54</b>	<b>48</b>	<b>89%</b>	

Table 1: Overall results.

The first thing to notice is that the classifier performs very well, 94% of all sentences and 89% of all lists are classified correctly. In Table 2 the results are presented for the various types of classifications.

The column “Found” shows the amount of sentences that have correctly been classified for a particular type. The column “Missed” gives the amount of sentences or lists that should have been classified as a particular type, but were not. One can for instance see that two sentences should have been classified as a ‘norm’ of type ‘right/permission’, but were not. The column ‘False’ presents the amount of sentences that were incorrectly classified as a particular type, e.g. in the same row as before three sentences were incorrectly classified as a ‘right/permission’ type of norm. Each false positive corresponds to a ‘missed’ somewhere else.

---

<sup>4</sup> <http://www.metalex.eu/>

<i>Type</i>	<i>% in corpus</i>	<i>Sentence</i>			<i>List</i>		
		Found	Missed	False	Found	Missed	False
Definitions	3%	10	1	-	3	-	-
Deeming Provision	0%	-	-	-	-	-	-
Norm – Right/Permission	16%	66	2	7	18	1	-
Norm – Obligation/Duty	6%	27	2	-	3	1	-
Application Provision	8%	40	-	5	3	-	1
Value Assignment	0%	1	-	-	-	-	-
Change	33%	168	6	6	-	-	-
Enactment Date	3%	13	2	-	-	-	-
Citation Title	0%	1	-	-	-	-	-
Penalization	0%	2	-	2	-	-	-
Mixed Type	0%	-	1	-	-	-	-
Norm – Statement of Fact (default)	31%	121	14	8	25	1(+3)	2
<b>Totals</b>	<b>100%</b>	<b>449</b>	<b>28</b>	<b>28</b>	<b>52</b>	<b>3(+3)</b>	<b>3</b>

Table 2: Results for types of sentences.

Norms obviously take the greatest part of all sentences in the laws we used: The explicitly recognized ones plus the default make out 58% of all phrases. The second largest category forms the so called ‘change’ class. These are the provisions that change some existing law; 33% of all sentences and lists belong to this class. The classifier even further specifies this type:

<i>Type</i>	Found	Missed	False
Change - Repealed	14	1	6
Change - Scope	41	-	-
Change - Replacement	65	3	-
Change - Renumber	6	2	-
Change - Insert	42	-	-
<b>Totals</b>	<b>168</b>	<b>6</b>	<b>6</b>

Table 3: Results for change provisions.

One sentence contained two changes: renumbering and a repeal. It is listed in Table 2 as ‘Mixed type’, and is not listed in Table 3. It used a specialised pattern that had not been added, rather than two patterns combined. (It may be better to classify this sentence as a separate type than to consider it a mixed type sentence.)

Five of the six false positives of the “repealed” type sentences were provisions concerning the repeal of fines instead of articles. This will require more sophisticated patterns or dedicated ‘anti-patterns’ (i.e. not applicable when it contains the word

‘fine’). The two false penalizations are in fact both a right<sup>5</sup>; the pattern that triggered this classification was part of a qualification of a legal body that was given certain rights. We will need a more sophisticated parser to detect this use of the pattern. The false rights and false application statements (twelve in total) have all been misclassified because the pattern for right or application statement appeared in an auxiliary sentence.

We only encountered one value assignments in the texts classified during this experiment. These seem to be specific to certain domains (i.e. taxes), and perhaps they are usually deferred to lower order regulations. We also found only one citation title and no deeming provisions at all.

Our assumption that all lists could be classified by their first sentence does not always hold. We found one curious exception in Bill 20585 nr. 2<sup>6</sup>:

Our Ministers:

- a. appoint, suspend and discharge the chairman and other members, after hearing the council involved;
- b. appoint, suspend and discharge the advising members.

This should perhaps have been written differently to start with: either not a list at all, or with the list starting after ‘discharge’. Three such lists, which could not be classified based on the header, were by the default classified as a statement of fact. This turned out to be the correct classification, but as they were not achieved in the correct manner, they have been counted as an error, and have counted a “missed” statement of fact in Table 2.

## Conclusions and Discussion

The classifier works very well, at least on the set of Dutch laws used in this experiment; 94% of all phrases was classified correctly and there were hardly any false positives. Almost 60% of all phrases was classified as some type of norm, a further 33% as clauses changing an existing law. A necessary pre-condition for the classifier is that the structure of the documents has already been marked. For most modern corpora this will not pose a problem, as they usually have been marked in such a way. For legacy texts, however, an automatic structure recogniser would be desirable. If the text is marked by hand, it is relatively easy to also classify it, and the gain from using the classifier will be rather small.

Despite these very positive results, there is of course room for improvement. An important threat to the accuracy of the classifier seems to be the occurrence of patterns in auxiliary sentences. Franssen [6] has suggested that this problem may be solved by a smart ordering of the patterns. By giving those patterns that may appear in the auxiliary sentences a lower priority, the chance is increased that a sentence is classified based on its principal sentence. However, this will still leave room for error. It would be preferable if somehow, the sentences in the input were split in principal and auxiliary sentences. This classification is merely the first step in a larger process to create a

---

<sup>5</sup> These are the two rights noted as “missed” in Table 2.

<sup>6</sup> Bill on Foreign, Development Cooperation and Defense Policy Councils (“Voorstel Raamwet Adviesraden buitenlands, ontwikkelingssamenwerkings- en defensiebeleid”).



model for these sentences, and the distinction between principal and auxiliary sentences will be of use in a later stage as well.

Another point of improvement is the granularity of the current classifier. With regard to the normative sentences, the classification is very coarse, with only two categories: Right/Permission and Obligation/Duty<sup>7</sup>. We suspect it will be possible to make more distinctions. Especially with regard to the norms of competence, it seems that there are several standardised constructs being used within the Dutch laws.

Will these results generalize to all Dutch law and possibly other jurisdictions? A pattern-matching approach often lacks generalization capabilities. Although languages do have underlying rules, people will often stretch and bend these to their need. This means that a system based on rules is often too rigid to deal with all the variation that can occur [8]. Therefore, a statistical approach is often advocated [8][9]. However, the amount of variation in legal texts is restricted, as legal drafters will seldom use a complete new style, instead using the style of older laws or the official guidelines. Our patterns were gathered from a completely different set of laws than the one we tested them on. The success rate of 94% strengthens us in the idea that it should generalize to all relatively new Dutch legislation. We will need to add patterns for older legislation and possibly for certain specific types like Tax Law. Likewise, there is no reason to assume similar success with different patterns could not be achieved for other jurisdictions. There too probably the language used in legislation will be restricted and contain typical patterns. In [10] an experiment is described for Italian law in which machine learning techniques (SVM) are used to classify paragraphs of law texts. They achieved an average of 90% correctness in classifying 582 paragraphs (provisions) into 11 types or classes. Their set of laws contained more ‘change’ type of sentences (50% as opposed to our 33%) and only 15% norms, but they do not mention the ‘statement of fact’ phrasing for normative expressions. They also have a large number of what they call ‘penalties’, our ‘penalization’ (20%) which leads us to suspect they used penal law as a domain.

Based on this other study it is tempting to conclude that a pattern based context free grammar works as good in classifying sentences in legislation as a machine learning approach. This is due to the simplicity and consistency in use of the patterns we found. In our experience, in the time needed to construct a training set for machine learning, probably all relevant patterns are already found. Of course, a definite conclusion cannot be drawn due to the difference between the domains of our study and the Italian study.

A future extension in our classification approach may be to use statistical data to choose between competing classifications. [11] suggest to move the other way, adding more linguistic data to improve a machine learning approach to classification (again using SVM). However, we will first see how far we can get by smart ordering and parsing techniques distinguishing principal and auxiliary sentences as described above.

## **Acknowledgements**

We would like to thank our student Gijs Kruitbosch for his work on the classifier and the anonymous reviewers for their comments on an earlier version of this paper.

---

<sup>7</sup> The “Statement of Fact” category displayed in the results also denotes obligations/duties.

## References

- [1] E. de Maat and R. Winkels. Categorisation of Norms. In: A. Lodder and L. Mommers (Eds.), *Legal Knowledge and Information Systems. Jurix 2007*, pages 79-88. IOS Press, 2007.
- [2] E. de Maat, R. Winkels and T. van Engers (in press). Making Sense of Legal Texts. To appear in G. Grewendorf & M. Rathert (eds), *Formal Linguistics and Law*. Mouton, De Gruyter, Berlin, 30 pages. Series for the volume Trends in Linguistics - Studies and Monographs (TiLSM).
- [3] E. de Maat, R. Winkels, and T. van Engers. Automated detection of reference structures in law. In T. van Engers (ed), *Proceedings of JURIX 2006*, pages 41-50, IOS Press, 2006.
- [4] E. de Maat. *Natural Legal Modelling*. Master's thesis, University of Twente, Enschede, 2003.
- [5] Aanwijzigingen voor de regelgeving. Circulaire van de Minister-President. Original in *Staatscourant 1992, 230*. Last modification in *Staatscourant 2005, 58*.
- [6] M. Franssen. Automated Detection of Norm Sentences in Laws. *Twente Student Conference on IT, 2007*.
- [7] T.M. van Engers, P.J.M. Kordelaar, J. Den Hartog, and E. Glassée. POWER: Programme for an ontology based working environment for modeling and use of regulations and legislation. In: Tjoa, Wagner and Al-Zobaidie, editors, *Proceedings of the 11th workshop on Databases and Expert Systems Applications* (IEEE), pages 327-334, Greenwich London, 2000.
- [8] C.D. Nanning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge (MA), 1999.
- [9] M.-F. Moens. Innovative techniques for legal text retrieval. *Artificial Intelligence and Law*, 9(1), pages 29-57, 2001.
- [10] C. Biagioli, E. Francesconi, A. Passerini, S. Montemagni and C. Soria. Automatic semantics extraction in law documents. In *Proceedings of the Tenth International Conference on Artificial Intelligence and Law (ICAIL '05)*, pages 133-140, ACM Press, New York, 2005.
- [11] T. Gonçalves, P. Quaresma. Is linguistic information relevant for the classification of legal texts? In *Proceedings of the Tenth International Conference on Artificial Intelligence and Law (ICAIL '05)*, pages 168-176, ACM Press, New York, 2005.