



UvA-DARE (Digital Academic Repository)

Towards a Legal Recommender System

Winkels, R.; Boer, A.; Vredereg, B.; van Someren, A.

DOI

[10.3233/978-1-61499-468-8-169](https://doi.org/10.3233/978-1-61499-468-8-169)

Publication date

2014

Document Version

Author accepted manuscript

Published in

Legal knowledge and information systems: JURIX 2014: the twenty-seventh annual conference

[Link to publication](#)

Citation for published version (APA):

Winkels, R., Boer, A., Vredereg, B., & van Someren, A. (2014). Towards a Legal Recommender System. In R. Hoekstra (Ed.), *Legal knowledge and information systems: JURIX 2014: the twenty-seventh annual conference* (pp. 169-178). (Frontiers in Artificial Intelligence and Applications; Vol. 271). IOS Press. <https://doi.org/10.3233/978-1-61499-468-8-169>

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (<https://dare.uva.nl>)

Towards a Legal Recommender System

Radboud WINKELS¹, Alexander BOER,
Bart VREDEBREGT & Alexander van SOMEREN
Leibniz Center for Law, University of Amsterdam

Abstract. In this paper we present the results of ongoing research aimed at a legal recommender system where users of a legislative portal receive suggestions of other relevant sources of law, given a focus document. We describe how we make references in case law to legislation explicit and machine readable, and how we use this information to adapt the suggestions of other relevant sources of law. We also describe an experiment in categorizing the references in case law, both by human experts and unsupervised machine learning. Results are tested in a prototype for Immigration Law.

Keywords. Semantic Web, citations, MetaLex, network analysis, cluster analysis, reasons for citing

1. Introduction

More and more sources of law are freely available online in the Netherlands, but also in the rest of Europe and the world. Most of the time however, these are stand-alone databases, containing one type of documents, not linked to other sources. For instance the Dutch portal for case law – rechtspraak.nl – contains a (small) part of all judicial decisions in the Netherlands. Case citations in these decisions are sometimes explicitly linked, references to legislation are not.²

From earlier research we know that professional users of legal documents would like to see and have easy access to related ones from other collections. E.g. when we evaluated a prototype system that recommends other relevant articles and laws to users of the official Dutch legislative portal, they told us they would like to see relevant case law and parliamentary information as well [11].

In this paper we present a first step in that direction. The new version of our portal presents relevant case law, given a legislative article in focus for a user, and adapts the ranking of relevant other articles based on the related case law. The idea is that judges in explaining and justifying their verdicts – applying the law in practice – indicate that the sources they cite are somehow related. For that reason, we also investigate the *reasons* for citing and whether these can be recognized automatically. If so, we could use these to even better suggest relevant sources to (professional) users. The ultimate aim of this research is to build a *Legal Recommender System* based on content filtering for now (see [9] for more on general recommender systems). Later on we may also include collaborative filtering, i.e. use the actions and behaviour of other users to predict relevant new material.

We will first describe how we created the network between case law and legislation, then the prototype recommender system for immigration law and next our attempt at

¹ Corresponding author: Radboud Winkels, Leibniz Center for Law, University of Amsterdam, PO Box 1030, 1000 BA Amsterdam, Netherlands; Email: winkels@uva.nl

² Except for a metadata element for recent cases in the header of the document that contains the ‘main’ article(s) for the decision.

classifying the use of references to legislation in court decisions. We will end with conclusions and a discussion of results.

2. Related Work

Several researchers have applied network analysis to legal data, but as far as we know only to one type of data at a time: case law as in [2][12], or legislation as in [4][6]. Van Opijnen [8] uses links to legislation in Dutch case law when deciding upon the relevance of a particular case, but not to suggest other relevant sources of law and not as an applied context for legislation. He distinguishes two types of references from case law to legislation: a procedural one and a substantive one. He is only interested in substantive ones. He also notes how often a decision references an article, the hierarchical position of the referred law and the document structure level of the reference (e.g. whether the reference is to a chapter or an article).

Zhang & Koppaka [13] discuss reasons for citing (RFC) prior cases in US case decisions: the text area around a case citation. Since the case in focus is citing another case, they compare this text area to the ones in the cited case. We are looking at references to legislation, so a simple text comparison of the two documents will not do.

3. Creating the Network

As stated above, the court decisions published at the official Dutch portal do not contain explicit, machine readable links to cited legislation. The texts are available in an XML format, basically divided in paragraphs using `<para>` tags, with a few metadata elements. The most relevant metadata for our purpose are:

- The date of the decision (*'Uitspraakdatum'*)
- The field of law (*'Rechtsgebied'*)
- The court (*'Instantie'*)

We decided to work with a subset of all available data and chose the field of immigration law. The area is large enough, has lots of (recent) cases and we have access to experts at the Dutch Immigration Service (IND). We used the 'field of law' metadata element mentioned above for the selection of cases. That resulted in a set of 13,311 documents to work with.

3.1. Locating and resolving references

For locating references to legislation we use regular expressions as we have done in the past together with a list of names and abbreviations of Dutch laws [5]. This list also contains the official identifier of the law (the BWB-number), which can be used for resolving the reference later on. We consider high precision to be more important than high recall. Users will forgive us if we miss a reference, but be annoyed by false ones.

We evaluated this procedure by checking 25 randomly selected documents by hand. These documents contained 163 references to legislation of which 141 were correctly identified (recall of 87%). There was one false positive (precision of 99%). The references we missed were mostly those to the *'Vreemdelingencirculaire'* (a lower law that has a different structure than regular ones) and treaties with very long names like *'Europees Verdrag tot bescherming van de rechten van de mens en de fundamentele vrijheden'*³. If we

³ 'Convention for the Protection of Human Rights and Fundamental Freedoms'.

tried to capture this with the regular expression, the regular expression would match too easily, often matching entire sentences where it should have matched only the law. We declared these conventions outside the scope of this experiment.

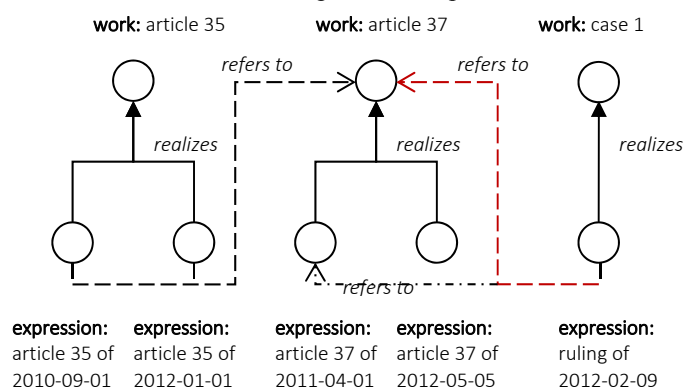


Figure 1: Bibliographic levels of documents and referencing. Article 35 has two expressions, both refer to the work level of article 37. A decision Case 1 (the only expression of the work) refers to article 37, given the date probably to the first expression, but we represent it as referring to the work level (red arrow).

Resolving the references was a bit trickier, since sometimes they used anaphora, e.g. referring to ‘that law’. In that case, the citation was resolved by using the previous law identifier if it existed. We used the same process for resolving ambiguous title abbreviations; e.g. ‘WAV’ is an abbreviation of ‘*Wet Arbeid Vreemdelingen*’, ‘*Wet Ammoniak en Veehouderij*’ and ‘*Wet Ambulancevervoer*’.⁴ Most of the time the full title is used before the abbreviation is used. Another issue is determining the exact *version* of the law the case refers to.⁵ Typically, a judge will refer to the version that is in force at the moment of the decision, but it may also be the version that was in force at the time of the relevant facts, or even sometimes an earlier version of the relevant law, etc. We cannot decide which version is the correct one without interpreting the content of the case. Therefore we decided to resolve the reference to the *work* level of the source of law, i.e. no particular version (cf. Figure 1). The resulting references are added to the XML of the case law document.⁶ The final network of the 13,311 case documents has 85,639 links to legislation (on average 6.5 references per case); the links connect the ECLI identifier⁷ of the case with the BWB identifier of the source of law (see above).

We evaluated the resolving process by checking 250 random ones of all the references found, by hand. Of these, 234 should have been resolved since the other 16 were outside of the scope of this experiment. 198 were resolved correctly (a recall of 85%). We had 10 ‘false’ positives, i.e. references that were declared out of scope, so a precision of 95%. The results were good enough to continue.

⁴ ‘Labour immigration law’, ‘Ammoniac and Livestock law’ and ‘Ambulance law’ respectively. Given our domain, the first one most likely is the correct one, but we want to implement generic mechanisms.

⁵ Which *expression* of the *work* in terms of bibliographic references as used by e.g. CEN MetaLex [10].

⁶ These references can be recognized by a ‘metaLexResourceIdentifier’ attribute of the ‘dterms:reference’ tag and the fact is has a ‘dterms:string’ as child.

⁷ ‘European Case Law Identifier’; see Council conclusions on ECLI at: [http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:52011XG0429\(01\)](http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX:52011XG0429(01))

3.2. Weighing the Network

Since case decisions may refer to the same source of law, e.g. an article, more than once, we count the number of references and compute the weight of the link between the case and the article as: $W = \frac{1}{n}$

Where n is the amount of occurrences of a certain reference and W is the weight of the edge. The lower the weight, the stronger the impact on the network is.

In [11] we described a first version of our legal recommender system. It only presented other legislative suggestions to a user, given her focus on a specific article. These suggestions were based on characteristics of the network of the tax laws used in the study. The system runs on top of the MetaLex document server [3], containing all Dutch regulations – including historic versions – from the official portal *wetten.nl* in linked data format. Now we include the network of cases referring to legislation as described above. All processing is done in real time.

4. Prototype Legal Recommender System

When a user clicks an article, the related case law and legislation is retrieved:

1. The system checks whether the article appears in the case law network. If so, it creates a so-called *ego graph*, a local network containing all the nodes and edges within a certain weighted distance from the current node [7]. Since the network only contains references from case law to legislation and within legislation, we know that if we take 2 steps – ignoring the direction of the reference – we will end up in a legislative node again. A weighted distance of 2 will give us most of the legislative nodes related to the current node, but even these networks may become rather large to handle for our present prototype, running on a simple machine.⁸ We start searching with a weighted distance of 0.4 and gradually increase it up to 2.0 until we have a sufficiently large, but still manageable network.

Figure 2: The prototype legal recommender system. The user has article 2a of the Immigration law in focus. Current version is January 2014 (see pull-down menu at left). Relevant other articles are presented in window A (red or dark border) on the left and below that relevant case law.

⁸ Single core virtual private server - notorious for low performance.

2. To find relevant legislation, the system also checks whether the current node is in the legislative network of the MetaLex document server [3]. If so, it again creates an ego graph, this time for an unweighted network. To control the size of the graph, we use only references coming from the selected version (expression) of the current node.
3. If we have two local networks, we want to combine them in order to (better) predict the importance of legislative nodes. To do this, we need to assign weights to the legislative graph. We chose the value 0.1 as it allows the legislative network to influence the result but not overrule the case law references.⁹
4. Finally, we use *betweenness centrality* on the combined network to determine the most relevant articles for the current focus. The betweenness centrality of a node is the sum of the fraction of all-pairs shortest paths that pass through that node. One would expect that the focus node has the highest betweenness centrality in the local network, but this proved not always to be the case. The focus node was however always in the top-5.¹⁰ The results are shown to the user in the top of frame A of **Figure 2**.

4.1. A First Evaluation

We asked several professional users of the Dutch Immigration Service to use the prototype system and fill in an evaluation form afterwards. Due to the holiday season, only 3 users replied so far, but they were positive. They appreciated the clean and uncluttered interface, indicated that it was easy to understand without help and liked the indication of the number of times a case was referring to an article when you click on a case. They also noted that the inclusion of case law added suggestions of relevant articles that were not available in the previous system. They complained about the slowness of the system and the fact that references to the ‘*Vreemdelingencirculaire*’ were missing (as we explained above).

Table 1: Ten most frequent patterns found by hand in a sample of 30 cases and automatically in all cases¹¹

	Dutch term	English translation	Sample	All
1	ingevolge	due to	58	28,609
2	als/in bedoeld(e) in/onder	as referred to in	58	21,564
3	op grond van	because of	32	17,574
4	met/om toepassing van/aan	pursuant to	29	15,574
5	(in...) (is) bepaald	is determined in	28	7,893
6	in/met strijd(ig) met	contrary to	17	5,793
7	in de zin van	in the sense of	13	3,607
	is niet van toepassing	shall not apply to	6	2,818
	schending van	violation of	4	2,607
8	zich/met beroep(en) op	to invoke	13	1,837
	gelet op	having regard to	4	1,765
9	krachtens	under	12	1,580
10	juncto/jo	in relation to	7	783

⁹ We use the NetworkX library in Python for this purpose.

¹⁰ In fact we use betweenness centrality approximation, which proved to use up to 50% less time without significant changes in the results.

¹¹ Actually number 10 (“juncto”) was lower down the list for all cases.

5. Reasons for Citing

We examined 30 randomly selected court decisions to identify repeating patterns used in citing legislation. The most frequent used terms were: “due to” (“*ingevolge*”) and “as referred to in” (“*als bedoeld in*”), cf. **Table 1**.

We hoped these keywords are somehow related to the reason of citing the legislation. To investigate whether this was the case, the matter was presented to three persons who are experts on this field: Two judges and one person who teaches students how to read court decisions. We had them sort cards on which the keywords were printed on one side and examples of their use in actual cases on the flip side.

The two judges were very sceptical and did not think that the keywords would be good indicators for categories. In fact they did not think that such categories existed at all. The third expert was very interested and came up with this categorisation:

1. Keywords that indicate a *selection*, identifying essential laws for a case. Examples from **Table 1** are: “pursuant to”, “because of”, and “to invoke”.
2. Keywords that indicate *application* of law. Examples are: “due to ...”, “as referred to in...” and “under”.
3. Keywords that indicate a *concluding* (denying) function; it is an answer to the first category. Examples are: “contrary to” and “by way of derogation from” (“*in afwijking van*”, not in **Table 1**).
4. The last category only consisted of the keyword “in relation to”. This is an arguably uninteresting category.

The first three categories coincide with steps in the task judges perform when deciding a case: *normative assessment* [1].

5.1. Assigning Keywords to References for Clustering

For all keywords occurring more than twice, a regular expression was constructed to automatically identify which keyword was present in a certain reference. These regular expressions are stored in a ‘csv’ file and are sorted for frequency like in **Table 1**. The algorithm processing the references found in the cases described above will greedily choose the first matching regular expression, without considering alternatives. This could result in a skewer distribution than actually is the case, but it is a relatively simple way to label the references. To counteract the unrealistic skewness, the first regular expressions have been made more complex to avoid false positives. In total 152,853 references were found in the domain of immigration law. Of those, 120,855 (79%) were assigned a keyword.

5.2. Other Features for Clustering

Apart from the keywords, some other features were identified:

Position of the reference in the document: The idea is that different sections of a decision play different roles. Though these sections are not distinguished in a machine readable form, their order remains more or less constant. Taking the relative position of the reference, these patterns should estimate in which section a reference is. In **Figure 3** the distribution of the variable is plotted for a sample. It is clear that the reference density is higher in the first section of a court decision.

Law-identification number: This feature is directly copied from the original xml data.

Type of court: This feature existed in two forms. One in which it is copied from the original xml file; the other is a simplified one that only distinguishes courts from higher

or supreme courts. This simpler version was used because some local courts were very rare.¹²

Year: The year of the decision, extracted from the original data. It could be that something changed in the reference-structure over the years. It was converted to ‘years ago’ by subtracting it from the current year.

Reference url: The actual reference, also copied from the original data. It is much more detailed than the law identification number above. Different sections of a law may play different roles, so this could very well be an important feature.

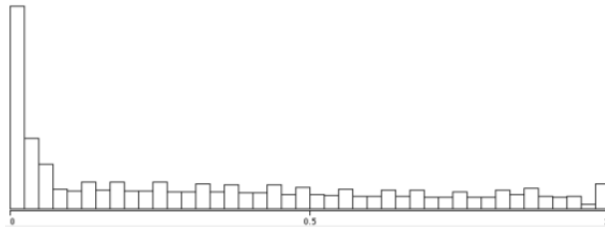


Figure 3: Distribution of the position of the reference in the court decision. A sample of 100,000 references was used for this plot.

5.3. Clustering References

Since we do not have an explicit theory for classifying references, nor a set of labelled data, we decided to apply unsupervised learning to see whether natural clusters exist in the data. Two rounds of analysis have been performed. The first round was more experimental, the second round was used for actual interpretation. In the first round all features mentioned were used except the reference URL. This feature was originally left out to drastically decrease the dimensionality of the input data. This made the results much easier to interpret. In the second round, the simpler version of the ‘type of court’ feature was used.

The clustering was performed in Weka 3.6.11¹³ using expectation maximisation (EM). This algorithm maximises the log likelihood of the data given the model by fitting Gaussian distributions on numeric attributes. It uses discrete estimators for nominal values. Since it is not clear how many clusters we are looking for, multiple numbers of clusters were tried and compared by cross validation. Higher numbers of clusters were penalising, since the log likelihood is guaranteed to increase in this algorithm when the number of clusters increases. This was performed using Weka by setting the `numCluster` variable to -1. The expectation maximisation algorithm has the advantage that the algorithm itself is comprehensible and its results are also easy to interpret. For the numeric attributes it will return mean (μ) and standard deviation (σ) of the best fitted Gaussian per cluster. For nominal values it will return frequencies that are Laplace corrected (i.e. added one to make dividing by zero impossible).

Every instance is part of every cluster with some likelihood. While these instances will eventually be assigned to the cluster with the highest likelihood for that instance, their values will be proportionally distributed according to the different likelihoods. Therefore all frequencies in the results will be greater than 1.

To evaluate how much attributes contributed to the formation of the clusters, the `ChiSquaredAttributeEval` of Weka was used. In this method, the χ^2 -statistic is calculated for every feature. The result is a rank list of the features, telling how much they contribute to distinguishing the classes.

¹² In fact, most cases were from courts in ‘Den Haag’.

¹³ See: <http://www.cs.waikato.ac.nz/ml/weka/>

5.4. Clustering Results

We used a sample size of 8,000 for the clustering experiment. One run took approximately ten minutes to complete. This made it possible to experiment with slight changes without it being too time consuming. Since the EM-algorithm is not guaranteed to find a global optimum, the experiment was conducted 5 times with different seeds (for the random initialisation component) and the results with the highest log-likelihood were used for interpretation. This still does not guarantee a global optimum, but this is a limitation of the algorithm that cannot be totally avoided.

From the results of the χ^2 evaluation (**Table 2**) it can be seen that ‘reference url’ is the most distinguishing attribute. This fits the idea that different parts of a law are referred to for different reasons.

The EM algorithm distinguished 12 different clusters (**Figure 4**). To be better able to interpret the results, the following steps were taken:

- All options of the attributes ‘reference url’ and ‘law identification number’ that occurred less than 12 times were ignored.¹⁴
- All frequency values (i.e. all except ‘position of reference’ and ‘year’) were normalized by dividing them by the relative frequency of the class and the total frequency of the attribute.

Cluster 3 is the largest and contains 30% of all instances. Almost all references (97%) are to the immigration law (‘*Vreemdelingenwet 2000*’) and 66% of all references to this law are in this cluster. There are no references to administrative law.

Table 2: χ^2 results of the final round EM clustering with 8,000 instances.

Average Merit		Attribute
μ	σ	
56,719	290	Reference url
28,381	188	Law identification number
12,786	46	Keyword
9,714	89	Position of the reference
1,478	25	Type of court
1,365	64	Year
0	0	Instance number

Cluster 1 contains relatively many references to a law concerning relief of asylum seekers (‘*Wet Centraal Orgaan opvang asielzoekers*’), but also many to immigration law. The references in this cluster appear at the beginning of a court decision.

Cluster 2 is difficult to categorize. It contains relatively many references with keywords like ‘due to’, ‘as referred to in’, ‘is determined in’ and variations. It may be the second category that our expert indicated as ‘applications of law’ (Section 5).

Cluster 7 contains all references to the Convention for the Protection of Human Rights and Fundamental Freedoms. Most of these references are to article 3, which concerns the prohibition of torture, article 8 (Right to respect for private and family life) and article 5 (Right to liberty and security).

Cluster 6 contains relatively many references to the administrative law articles 8.81-8.87 on preliminary provisions (‘*voorlopige voorziening*’).

¹⁴ An arguably arbitrary number based on the number of clusters (12).

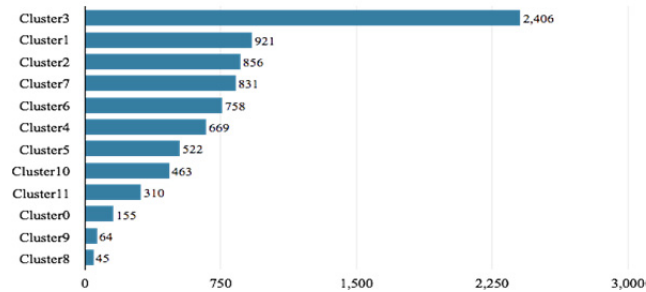


Figure 4: Frequency distribution of the clusters found with the EM-algorithm with 8,000 instances.

Cluster 4 contains relatively many references to the Administrative Law, mostly of a procedural kind. Just as cluster 1, most occur at the beginning of the decision.

Cluster 5 contains quite a number of references to the Convention on the Rights of the Child. Contrary to clusters 4 and 1, most references in this cluster occur at the end of decisions.

Of the remaining smaller clusters it is interesting to note that **Cluster 11** contains only very recent cases (μ 'Year'=1.7 years from 2014 with $\sigma=0.82$). **Cluster 0** only contains references with the keyword 'does not apply' together with art. 6.6 of the Administrative law, which is about whether an appeal can be declared inadmissible. **Cluster 8** finally is a very small, but interesting cluster that contains references from an average 13.3 years ago (with a standard deviation of 1.05 years). Note that 14 years ago a new immigration law was adopted. Maybe that considerably changed something in the data as a result of which the EM algorithm needed to assign a separate cluster to it.

6. Conclusions and Further Research

We have shown that it works quite well to automatically find and resolve references to legislation in Dutch case law. These references can be used to provide users of the legislative portal with relevant judicial decisions given their current focus and moreover, suggest additional relevant legislative sources. The parser can easily be improved a little and the prototype system will perform much better when running on a proper server. We also did not exploit the network of case law itself this time. In [12] we showed that this can be used to estimate the authority of cases, so if we include this the suggestions of relevant case law should be improved.

We also showed that it is possible to categorise references from case law to legislation. Literature and experts did not really give us an indication of the categories we needed to look for, therefore we used a data driven approach. That way we could distinguish 12 clusters as described above. Most of the clusters make sense and are domain specific; the specific (part of) law that is cited is the main distinguishing feature. The keywords we had our experts sort are less important. Maybe only cluster 2 is an example of a more generic cluster that was also indicated by one of our experts. It is important to keep in mind that a data driven categorisation does not have to be natural or acceptable for legal professionals. The question of whether this categorisation is useful for them remains open.

In the future we may also decide to use some additional features as:

- The relative frequency of the reference in a court decision (what van Opijnen called 'multiplicity' [8]).
- The hierarchical position of the law, e.g. whether the referred law is a European directive or treaty, or a governmental decree.

- Document structure level. A lower document structure level (e.g. article or clause instead of a chapter) suggests a more specific reference, which could indicate a different role.
- Bag of Words, which was actually implemented, but left out due to increased complexity.¹⁵ The bag of words could be interesting on both sides of the reference. It could be performed on both the paragraph of the court decision in which the reference occurs, but also on the section of the law the reference refers to.
- The position of the specific section referred to. The idea is similar to the used position of the reference in the case decision. Laws are drafted in a structured way. The position could indicate the role an article plays in the law and thus a certain role of the reference in the court opinion. An example is that definitions often appear in the beginning of a law.
- We also intend to use some other data we extract from the cases: An ordered list of all the references in the decision. We are interested whether there is structure in these lists. Do some references always occur before others, etc. Another use of these data is creating a list of n -grams to help predict the next references. E.g. if we have a set of bi-grams and a previous reference we could look up in our bi-gram set which reference is most likely to come next, enabling us to create an autocomplete system for references.

Acknowledgements

Part of this research is co-funded by the Civil Justice Programme of the European Union in the OpenLaws.eu project under grant JUST/2013/JCIV/AG/4562.

We would like to thank the legal experts for their contribution and the people of the Dutch Immigration and Naturalisation Service for evaluating the prototype.

References

- [1] Breuker, J.A. (1993). *Modelling Artificial Legal Reasoning*. Knowledge Acquisition for Knowledge-Based Systems, LNCS Volume 723, 1993, Springer, Berlin, pp. 66-78.
- [2] Fowler, J.H., Johnson, T.R., Jeon, S. and Wahlbeck, P.J. (2006). Network analysis and the law: Measuring the legal importance of supreme court precedents. *Political Analysis*, 15(3):324–346.
- [3] Hoekstra, R. (2011). The MetaLex Document Server - Legal Documents as Versioned Linked Data, *Proceedings of the International Semantic Web Conference (ISWC2011)*, pp. 128-143. Springer, Berlin.
- [4] Liiv, I., Vedeshin, A. and Täks, E. (2007). Visualization and structure analysis of legislative acts: a case study on the law of obligations. *ICAIL 2007*, pp. 189-190, ACM.
- [5] Maat, E. de, Winkels, R., and Engers, T. van (2006). Automated detection of reference structures in law. In T. van Engers (ed), *Legal Knowledge and Information Systems. JURIX 2006*, IOS Press, Amsterdam, pp. 41-50.
- [6] Mazzega, P., Bourcier, D. and Boulet, R. (2009). The network of French legal codes. In *ICAIL 2009*, pages 236–237.
- [7] Newman, M. (2010). *Networks: An Introduction*. Oxford, England: Oxford University Press.
- [8] Opijnen, M. van (2014). *Op en in het Web*. Boom Juridische Uitgevers, Den Haag (in Dutch).
- [9] Ricci, F., Rokach, L., Shapira, B. and Kantor, P. (eds) (2011). *Recommender Systems Handbook*. Springer Science & Business Media, LLC 2011.
- [10] Saur, K. G. (1998). Functional requirements for bibliographic records. *UBCIM Publications - IFLA Section on Cataloguing*, 19:136.
- [11] Winkels, R.G.F., Boer, A. and I. Plantevin (2013). Creating Context Networks in Dutch Legislation. In K. Ashley (ed). *Legal Knowledge and Information Systems. JURIX 2013*. IOS Press, Amsterdam, pp. 155-164.
- [12] Winkels, R.G.F., de Ruyter, J. and Kroese, H. (2011). Determining Authority of Dutch Case Law. In K. Atkinson (ed). *Legal Knowledge and Information Systems. JURIX 2011*. IOS Press, Amsterdam, pp. 103-112.
- [13] Zhang, P. & Koppaka, L. (2007). Semantics-based legal citation network. *ICAIL 2007*, pp. 123-130, ACM.

¹⁵ Execution time of the EM algorithm with bag-of-words was more than 24 hours.