Text-mining for Lawyers: How Machine Learning Techniques Can Advance our Understanding of Legal Discourse

Arthur Dyevre*

Abstract

Many questions facing legal scholars and practitioners can be answered only by analysing and interrogating large collections of legal documents: statutes, treaties, judicial decisions and law review articles. I survey a range of novel techniques in machine learning and natural language processing – including topic modelling, word embeddings and transfer learning – that can be applied to the large-scale investigation of legal texts

Keywords: text mining, machine learning, law, natural language processing

1 Introduction

Much of the information of interest to lawyers and legal scholars comes in the form of texts. Whether they are briefs, contracts, court rulings, law review articles, legislative acts, treaties, newspapers or blog posts, all are either legal documents themselves or documents about the law. Retrieving, analysing, commenting, relating and expounding these documents has been the bread and butter of legal practice and legal scholarship alike for centuries.

Lawyers deal in words, and the law can be viewed as a vast and complex network of interrelated texts, as illustrated in Figure 1. The function of this discourse is not only to announce legal rules and how they apply to a particular set of facts but also to explain or summarise them in more succinct or more accessible language – which is understood to be one of the core functions of traditional, doctrinal scholarship.

While the study of legal texts is at least as old as academic legal scholarship, what is new is that a whole range of text mining techniques have emerged to assist the legal community in navigating and analysing the ever-expanding sea of legal and law-related documents. These techniques rely on recent advances in machine learning and natural language processing.

* Arthur Dyevre is Professor at the KU Leuven Centre for Empirical Jurisprudence, Leuven, Belgium. arthur.dyevre@kuleuven.be. I am grateful to Dr. Nicolas Lampach, Dr. Timothy Yu-Cheung Yeung, Monika Glavina, Kyra Wigard and Nusret Ipek for their invaluable research assistance. I acknowledge financial support from European Research Council Horizon 2020 Starting Grant #638154 (EUTHORITY).

The media hype about artificial intelligence (AI) occasionally leads to exaggerated claims about the capabilities of these techniques. Except for the simplest legal tasks, robot lawyers are not yet around the corner. Nor are fully automated robot judges (provided that robot judges are even desirable, which is, at least, questionable). However, even if the media hype (sometimes amplified by legal scholars) paints a misleading picture of what AI can achieve, it would be at least equally wrong to dismiss these techniques as irrelevant to legal practice or legal scholarship. This is true even for those who see themselves as hardcore black-letter law scholars. The now famous Gartner Hype Cycles tell us that perceptions of AI advances oscillate between peaks of inflated expectations and troughs of disillusionment before reaching a plateau of productivity.[1]

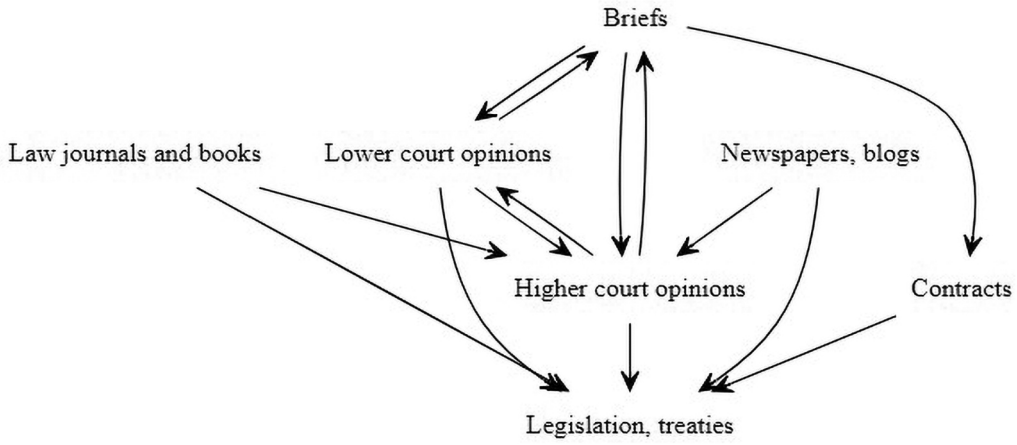
Researchers with experience in text-mining applications in the legal domain recognise that text-mining techniques cannot (yet) fully replace careful human reading. Yet these technologies are already sufficiently mature and progressing at a breakneck pace to deliver substantial advances. While increasingly popular in the interdisciplinary fields of law and economics, empirical legal studies and law and politics,[2] text-mining methods are also directly relevant to the work of doctrinal legal scholars. Indeed, one way to view them is as augmented doctrinal reality.

The present contribution aims to introduce these techniques to jurists who are unfamiliar with machine learning and natural language processing or who may only have a faint notion of the use that these tools can be put to. To this end, I shall first describe how data-harvesting methods can be deployed to gather large collections of legal documents. I will then proceed to explain how text is transformed into input data for text-mining tasks. Next, I will offer an overview of the text-mining techniques themselves, distinguishing supervised and unsupervised methods and walking the reader through a

1. *See* www.gartner.com/smarterwithgartner/5-trends-drive-the-gartner-hype-cycle-for-emerging-technologies-2020/ (last visited 2 March 2021).
2. For a review *see* J. Frankenreiter and M.A. Livermore, 'Computational Methods in Legal Analysis', 16 *Annual Review of Law and Social Science* 39-57 (2020); for reflections and illustrations of the use of machine learning and natural language processing methods in empirical legal studies *see* M.A. Livermore and D.N. Rockmore, *Law as Data: Computation, Text, & the Future of Legal Analysis* (2019).

doi: 10.5553/ELR.000191 - ELR 2021 | No. 1

Figure 1 *Law as text*



bunch of examples from the EUTHORITY Project (www.euthority.eu). Finally, because I expect that many readers might be interested in learning some of the reviewed techniques, I will say a few words about practical software implementation.

The present article addresses mainly continental European legal scholars. Because of this deliberate focus, the discussion deliberately excludes tasks and questions – such as contract review or document assembly – that are important in legal practice,[3] but of lesser relevance to academic legal research, as traditionally understood in continental Europe. Nor do I engage matters such as causal inference that are central to the integration of text-mining and machine learning approaches in empirical legal studies and law and economics.[4] Furthermore, my aim is to introduce text-mining methods in terms that my target audience (hopefully) will find understandable. For this reason I eschew mathematical notation and technical jargon to focus on the underlying conceptual intuitions with the help of concrete illustrations. Obviously, this comes at the cost of precision. But I hope that this sacrifice earns the benefit of lowering the barrier to access. It is also worth mentioning at the outset that the scope of the present review is, by its very nature, limited. Text-mining and natural language processing have become vast fields, currently progressing at a breakneck pace possibly unmatched in any other field of scientific inquiry. So to pretend that this survey is, in any sense, comprehensive would be silly.

The present contribution assumes that legal scholars, with or without prior training in statistics or empirical methods, can become not just intelligent consumers but also active users of this panoply of powerful techniques. Readers interested in applying computational textual methods will find some pointers in the section on 'Learning Text-Mining Methods'.

3. Efforts to automate these tasks have been an important focus of the emerging Legal Tech scene, *see* R. Dale, 'Law and Word Order: NLP in Legal Tech', 25 *Natural Language Engineering* 211-17 (2019).
4. *See* Livermore and Rockmore, above n. 3.

2 Harvesting Legal Texts

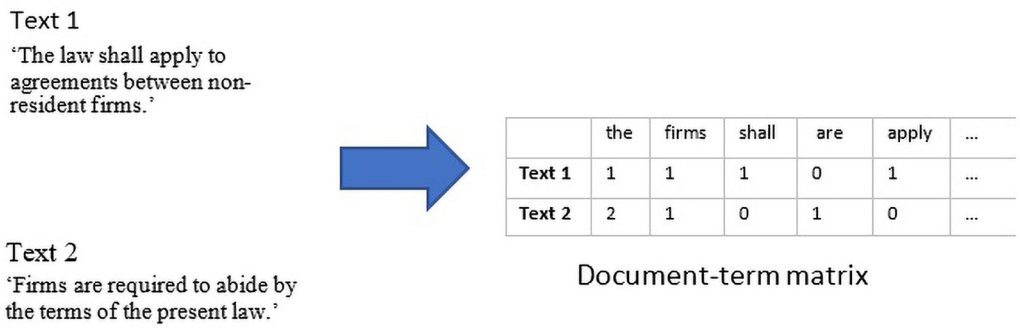
Computerised text-mining methods require that texts be in digital form. Luckily, millions of legal documents are now available at a few clicks in electronic repositories and legal databases. The degree of exhaustiveness of these repositories varies widely from jurisdiction to jurisdiction. At best, judicial databases offer access to all published decisions. Often, it will only be to a subset of these decisions, with older rulings typically less likely to make the cut. Because the universe of documents is somewhat smaller, legislative databases usually fare better, although, here too, there are jurisdictional and cross-national disparities.[5] As official gazettes are increasingly published digitally, they potentially represent a treasure trove of legal data.

When documents are not available in digital format, it is still possible to convert them to this format using scanning combined with Optical Character Recognition (OCR). OCR works better with more recent, undamaged texts than with old dusty casebooks or well-worn legal treatises. However, the technology has made huge strides, thanks mainly to machine learning (which helps guess semi-erased words or phrases). It is now even possible to digitalise handwritten documents,[6] opening up new possibilities for legal historians to scour old manuscripts.

When done manually, assembling a large collection of legal documents for a text-mining project can be excruciatingly time-consuming (try to download all European Court of Justice decisions since 1954). However, data-harvesting techniques can make this step considerably easier. Using libraries designed for this purpose in popular programming languages like R and Python, it is possible to download the entire content of EUR-Lex (the EU law database) with less than five lines of code.

5. At the European Union level, EUR-Lex is fairly comprehensive with regard to both legislative acts and case law. EU law – EUR-Lex, https://eur-lex.europa.eu/homepage.html (last visited 9 November 2020). National databases are typically less complete.
6. Digitize Your Notes With Microsoft Computer Vision API | Nordic APIs |, Nordic APIs (2017), https://nordicapis.com/digitize-your-notes-with-microsoft-vision-api/ (last visited 9 November 2020).

Figure 2 *Converting text into document-term matrix*



	the	firms	shall	are	apply	...
Text 1	1	1	1	0	1	...
Text 2	2	1	0	1	0	...

Web scraping, as the method is commonly referred to, is now the main go-to technique for collecting data in social scientific disciplines.[7] As scientists in various fields, including physics and medicine, have turned to text-mining methods to summarise vast collections of peer-reviewed articles, publishers (notably Oxford University Press and Elsevier) have made their journal collections available. Note, though, that the terms and conditions of commercial and non-commercial databases may sometimes explicitly prohibit web scraping, while there remain some uncertainties about when web scraping may be prohibited even for non-profit, purely academic research purposes.

3 From Text to Data

When we read a text our brain parses it applying our knowledge of semantics, syntax and context. In any language, the stock of words is finite, but syntactic rules allow the construction of infinitely many sentences from this finite vocabulary. Moreover, humans are able to communicate more than they say or write by taking the context into account. This is why we ascribe different meanings to the sentence 'I would like a table' when uttered in a restaurant and when uttered in a furniture shop.[8] While the language of legal documents – including contracts, statutes and judicial opinions – can diverge, sometimes significantly ('Any proviso to the contrary notwithstanding'), from everyday language, these basic principles of linguistic cognition and interpersonal communication are equally valid in the legal domain as in other areas of human activity.

Text-mining methods do not parse texts quite the same way the human brain does. Instead, these methods typically involve a good deal of complexity reduction. This may seem surprising to those less well-versed in machine learning. But even the most advanced natural language processing algorithms are still based on statistical principles. Texts are represented as numbers, in which the algorithms look for patterns. The ability to detect patterns depends on the amount of textual data and the sophistication of the algorithm, but the basic principle remains the same, including for the most cutting-edge techniques. In that sense, it is not entirely wrong to say that machine learning algorithms are still quite dumb. Yet their power stems from their ability to leverage the brute force of computing to arrive at useful (and sometimes surprisingly good) approximations.

Until recently, most text-mining methods relied on what is known as the bag-of-words (BOW) approach. To see what this amounts to, let us assume that we have a corpus with two texts, Text 1 and Text 2, as in Figure 2. The BOW approach involves converting texts into sequences of word counts and corpora to document-term matrices. The sequence of word counts representing a text is called a 'vector'. This vector contains counts of all the words occurring in that text and zeros for the words occurring in the other texts but not in that particular text. For Text 1 the zeros will represent all the words that appear in Text 2 but not in Text 1 and vice-versa. In a large corpus spanning a vocabulary of millions of words, the vector of word counts representing a text will contain mostly zeros – accounting for all the words that occur in other texts but not in the one under consideration.

To keep some phrases such as 'European Union' or 'Court of Justice' together instead of treating their component words as distinct lexemes, it is possible to throw some bigrams or trigrams into the document-term matrix. Think of an n-gram as a contiguous sequence of words. A bigram is a sequence of two words; a trigram a sequence of three words, and so on. Turned into a bigram 'European Union', for example, becomes 'European_Union', whereas 'Court of Justice' becomes the trigram 'Court_of_Justice'. These n-grams can then be processed just as individual words (unigrams).

In many applications, it is also common to remove so-called 'stopwords' – articles and prepositions like 'the', 'but', 'to', etc. – and to convert all words to lower case.[9] The resulting document-term matrix is the basic input

7. N.J. DeVito, G.C. Richards and P. Inglesby, 'How We Learnt to Stop Worrying and Love Web Scraping', 585 *Nature* 621-2 (2020).
8. D. Sperber and D. Wilson, *Relevance: Communication and Cognition* (1996).
9. Some text-mining tasks such as authorship identification require a distinct approach to pre-processing. Indeed, because pronouns and prepositions are markers of personal style, it is common to restrict the document-matrix to this class of words and to exclude nouns, verbs and adjectives.

doi: 10.5553/ELR.000191 - ELR 2021 | No. 1

of many popular text-mining methods, such as latent semantic analysis (LSA) or topic modelling.

This modus operandi may strike many as a crude simplification. Yet, crude as it may be, this simplification can nonetheless produce useful results, as we shall see.

It is easy to see, however, that progress in modelling language and improvements in the performance of downstream applications – in law just as in other fields – ultimately entailed bringing the field beyond the BOW paradigm to develop richer representations of vocabularies while capturing more of the context and rules of syntax.

As we will see, static word embedding models such as Word2Vec have taken a significant step in that direction by representing words by their co-occurrence associations. These methods reflect the emergence of new paradigm building on notions from distributional linguistics, notably the intuition that a word is defined by the company it keeps.

Cutting-edge methods like transformers have taken the field several steps further into this new paradigm. Pre-trained on giant corpora, transformer models like Google's BERT (Bi-directional Encoder Representation Transformer) rely on a contextualised representation of word usage, enabling them to handle polysemy and to parse the reference of pronouns – a remarkable achievement that constitutes a major milestone in the development of AI language models.

Note, however, that while these novel techniques do not require converting raw texts to a document-term matrix, they still require texts to be in digitalised, machine-readable format.

4 Unsupervised Techniques

Computer scientists and machine learning scholars typically speak in terms of tasks –information retrieval, clustering, summarising, forecasting, etc. – or in terms of whether the method or algorithm operates with human-labelled documents or not – supervised versus unsupervised.

Translated into more familiar language, information retrieval is what jurists do when they search a document collection for a specific set of documents: e.g. entering a list of keywords into a database search engine to retrieve all judicial rulings addressing a particular issue. Similarly, clustering is what lawyers do when they try to sort out documents into categories: e.g. the themes to which law review articles relate or the topics coming up in judicial rulings. Turning long documents into more easily digestible summaries is also something that lawyers do on a routine basis. Prediction is something that one may not intuitively associate with texts. Yet words, too, whether from legal briefs or other textual inputs, can also serve to predict events or behaviours.

Techniques referred to as 'supervised' are those that necessitate human-labelled documents. They operate by seeking patterns correlated with human annotations, and their ability to predict how humans would annotate unlabelled documents is the measure of their performance. 'Unsupervised' techniques, on the other hand, do not require manually labelled textual input. However, the output they generate requires human interpretation or validation.

Some techniques and machine learning algorithms have been specifically designed for particular tasks. Yet several methods, some supervised, others unsupervised, may sometimes come into consideration for the same task, in which case the optimal choice should ultimately depend on the specific research question of interest to the legal analyst.

4.1 Word Cloud

Word cloud plots are arguably one of the most familiar and simplest text-mining methods. A word cloud simply plots words according to their aggregate frequency in the document-term matrix. Illustrated in Figure 3 is a slightly more sophisticated word cloud, known as a 'comparison cloud'.[10] It is based on a corpus compiling all European Court of Justice rulings up to 2015 (over 12,000 documents). Plotted are not the most frequent words in the overall corpus but the words that are most distinctive of the three main procedures: annulments (Art. 263 Treaty on the Functioning of the European Union (TFEU)); infringements (Art. 258 TFEU) and preliminary rulings (Art. 267 TFEU).

Our comparison cloud suggests that 'undertakings' is more distinctive of annulment proceedings (maybe because European Commission competition decisions reach the Court via this procedural channel), whereas 'agreement' and 'sugar' are more characteristic of, respectively, infringement and preliminary rulings.

Word clouds are popular and easy to interpret but are rather crude tools when it comes to detecting more granular patterns. In some applications pre-processing steps, such as restricting the document-term matrix to certain parts of speech (e.g. nouns or adjectives) may help make them more informative. But limitations remain.

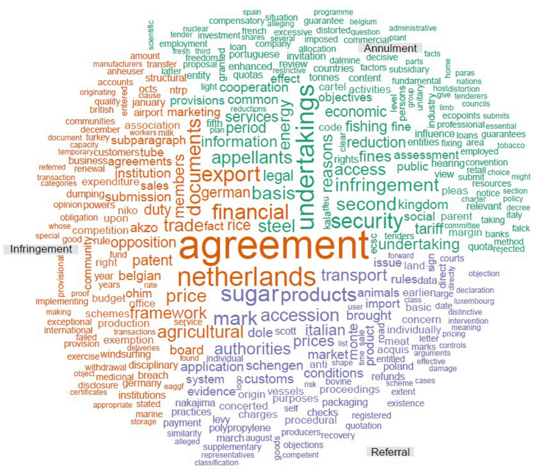
4.2 Latent Semantic Analysis and Principal Component Analysis

A notch more advanced are principal component analysis (PCA) and latent semantic analysis (LSA). Both are closely related and relatively old statistical techniques to arrange large arrays of data into more interpretable patterns. In the field of text mining, they fundamentally serve as unsupervised clustering methods to explore how texts and their words relate to each other.

PCA and LSA both work by seeking to represent the high-dimensional variations in word usage – a corpus and the documents it comprises vary in as many ways as the number of words in its vocabulary – into something

10. The size of a word reflects its deviation from their average across documents. Suppose p_{ij} is the rate at which word i occurs in document j and p_i its average rate across documents ($\frac{\sum p_{ij}}{number\ of\ documents}$). Word size is determined by the maximum deviation ($max(p_{ij} - p_i)$).

Figure 3 Comparison word cloud of infringement, annulment and preliminary rulings



more easily interpretable (and cognitively manageable) for the human brain. The output of both statistical procedures are a smaller number of dimensions on which words and documents are arrayed to facilitate the identification of meaningful patterns of relatedness.

The patterns of interest and the words expressing them depend on the specific task. PCA, for example, has been used to identify the authorship of *The Federalist Papers*.[11] But both methods can also be used to cluster legal documents around themes if one of the generated dimensions allows such an interpretation.

Figures 4 and 5 illustrate the use of LSA to explore oscillations in the position of the German Federal Constitutional Court over European from the 1960s and up to 2020. The corpus comprises 26 rulings, whose lengths vary from a little more than 1,000 to more than 20,000 token words.[12]

Here the basic interpretive assumption with which the output of the algorithm was approached is that variations in jurisprudential stance should be reflected in the use of words related to statehood and the internal market, with greater divergence in vocabulary manifesting greater jurisprudential divergence.

Figure 4 shows the extent to which selected clusters of words tend to appear in the same decisions. Unlike in word cloud plots such as the one depicted in Figure 3, the position of words has a precise meaning here. Vertical and horizontal axes denote separate dimensions, while the position of words is itself related to the documents in which they occur. If two documents share many words that are close to each other on a dimension, these documents will also be close to each other on that particular dimension. For example, 'sovereignty of the people' (*Volkssouveränität*), 'constitutional identity' (*Verfassungsidentität*), 'enumerated powers' (*Einzelermächtigung*) and '*ultra vires'* are close to each other on Dimension 1. These words are also more closely associated with the Court's more Eurosceptic judgments, like *Maastricht* and *Lisbon*. 'Duty to refer' (*Vorlagepflicht*), 'direct' (*unmittelbar*), 'effect' (*Wirkung*), 'export' (*Ausfuhr*), 'good' (*Ware*) form another separate cluster on the same dimension on the right-hand side. These words are also more closely associated with integration-friendly rulings, like *Kloppenburg*, *Banana* or *Lütticke*.

If we interpret Dimension 1 as Europhilia, the document positions associated with Dimension 1 can be interpreted as capturing the rulings' expressed position over European integration. Figure 5 depicts document positions on Dimension 1 over time. It shows that the resulting scaling is highly consonant with the conventional doctrinal wisdom. The *ECB Ultra Vires* ruling, in which the German Court declared the Court of Justice decision in *Weiss* ultra vires, clearly scores as the most Eurosceptic ever. Other rulings, such as *Maastricht*, *Lisbon* and *OMT*, which borrow the same state-centric sovereignty rhetoric, are also on the more Eurosceptic side, in keeping with the conventional wisdom.

A recent article compared the performance of eight algorithms, including LSA, in mapping the evolution of the German Court's case on European integration. The positions ascribed to the decisions by the algorithms were evaluated against scholarly accounts and legal expert ratings. A variant of LSA (correspondence analysis) performed best against scholarly accounts in law journals, achieving a 75% pairwise correlation.[13]

4.3 Topic Modelling

A more recent technique specifically designed for clustering and automated classification is topic modelling.[14] Suppose you have a large amount of legal texts and you want to get a sense of the themes and topics they pertain to. Instead of asking you to come up with a list of cate-

11. D.I. Holmes, 'Authorship Attribution', 28 *Computers and the Humanities* 87-106 (1994).
12. For a discussion and assessment of the performance of LSA and other text-mining methods to map jurisprudential change, *see* A. Dyevre, 'The Promise and Pitfall of Automated Text-Scaling Techniques for the Analysis of Jurisprudential Change', *Artificial Intelligence and Law* 1-31 (2020).
13. *Id.*
14. For a non-technical introduction *see* D.M. Blei, 'Probabilistic Topic Models', 55 *Communications of the ACM* 77–84 (2012).

Figure 4 *Frames and phraseology of German constitutional rulings on Europe*

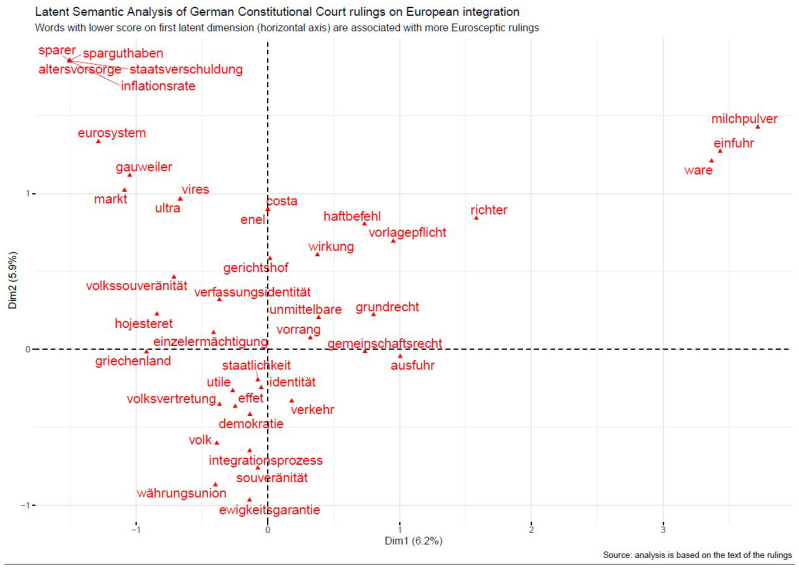
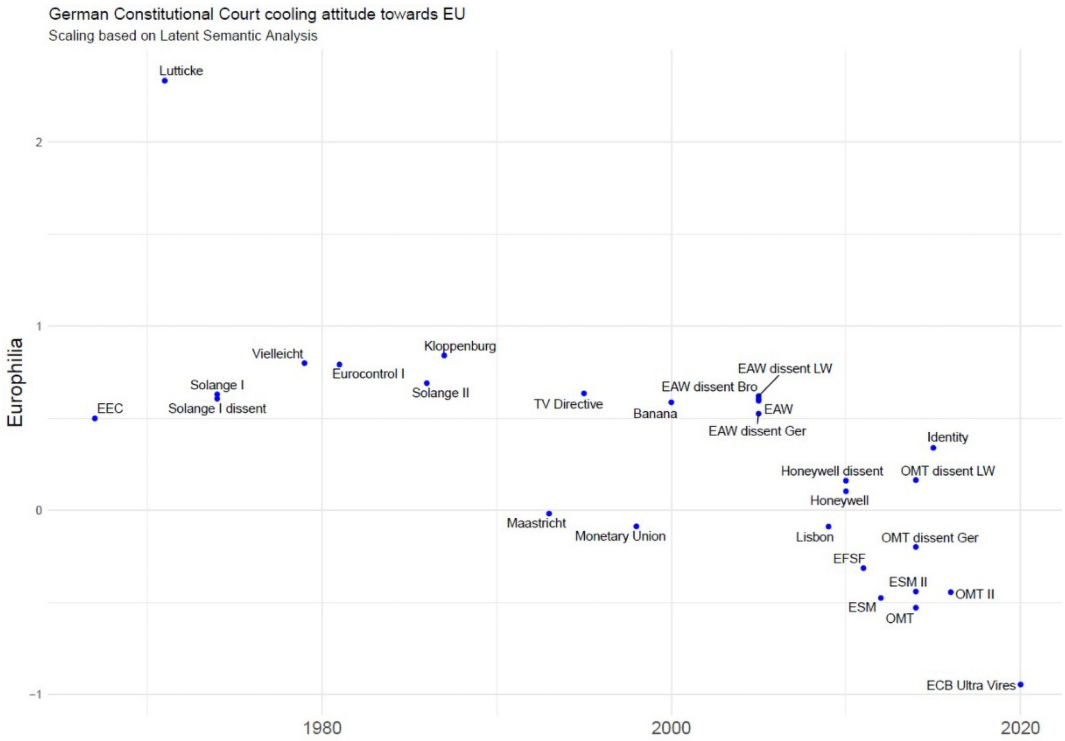


Figure 5 *Evolution of the German Constitutional Court's stance on European integration based on Dimension 1 of LSA*



gories or a classificatory scheme, topic modelling generates the categories and sorts out the documents accordingly after you have specified how many topics you wanted. At least, this is how the method is supposed to work.

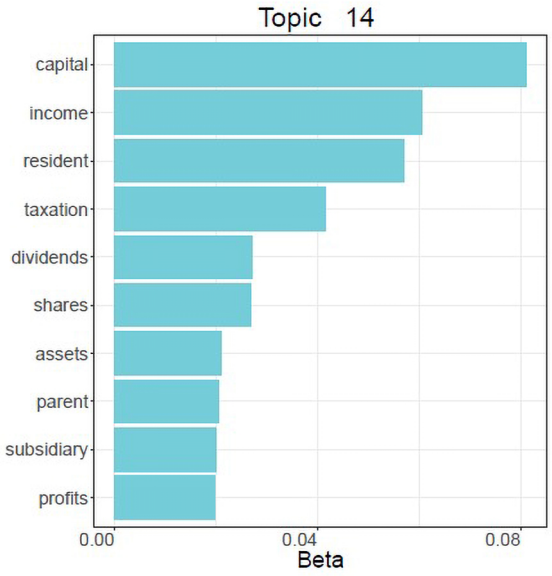
In topic modelling, topics are modelled as probability over words and documents as probability over topics. To generate the topics, the algorithm tries to find which probabilities are most likely to have generated the observed documents.

Figure 6 illustrates the output of a topic model of preliminary rulings (approximately 8,000 rulings). The number of topics was set at 25. What Figure 6 displays is one of these topics represented by its 10 most distinctive words (note that the higher the beta value, the more characteristic of the topic the word is). Looking at these 'keywords' – which, it is essential to understand, are not chosen by the researcher but emerge from the analysis – we may plausibly summarise this topic as corporate taxation.

In topic modelling, documents are conceptualised as mixtures of topics and, in addition to generating topics, a topic model tells you what proportion of what topic documents are likely to contain. So to check that our interpretation of topic 14 is correct we can inspect the decision that, according to the model, has the highest proportion of this topic. In that case, it turns out to be *Test Claimants in the FII Group Litigation v Commissioner of Inland Revenue*[15], a 2012 Grand Chamber ruling,

15. 12 December 2012, C-446/04.

Figure 6 Topic from topic model of preliminary rulings (1961-2016)



which according to the model is 99% about topic 14. Here is a quote from the first ruling:

> The High Court of Justice of England and Wales, Chancery Division, seeks, first, to obtain clarification regarding paragraph 56 of the judgment in Test Claimants in the FII Group Litigation and point 1 of its operative part. It recalls that the Court of Justice held, in paragraphs 48 to 53, 57 and 60 of that judgment, that national legislation which applies the exemption method to nationally-sourced dividends and the imputation method to foreign-sourced dividends is not contrary to Articles 49 TFEU and 63 TFEU, provided that the tax rate applied to foreign-sourced dividends is not higher than the rate applied to nationally-sourced dividends and that the tax credit is at least equal to the amount paid in the Member State of the company making the distribution, up to the limit of the tax charged in the Member State of the company receiving the dividends.

So it does really look like corporate taxation after all.

Topics can be visualised in various ways. In Figure 7, they are represented as a network in which node size represents overall topic proportion in the overall document collection while edge thickness corresponds to the weighted number of shared words. This way we can see themes emerging from the topics.

Among other things, Figure 7 suggests that internal market and tax issues represent a big chunk of what the Court of Justice of the European Union (CJEU) does. However, social rights, residence rights and the recognition of foreign judgments (private international law) also make for a substantive share of the cases on which the Luxembourg judges sit.

If you think that 25 categories is too few to get a good sense of issue prevalence in the Court's case law, how about a topic model with 100 categories? In Figure 5 we see that such a topic provides a more detailed picture, although we find the same themes (internal market in the lower-right region, social and immigration issues in the left region).

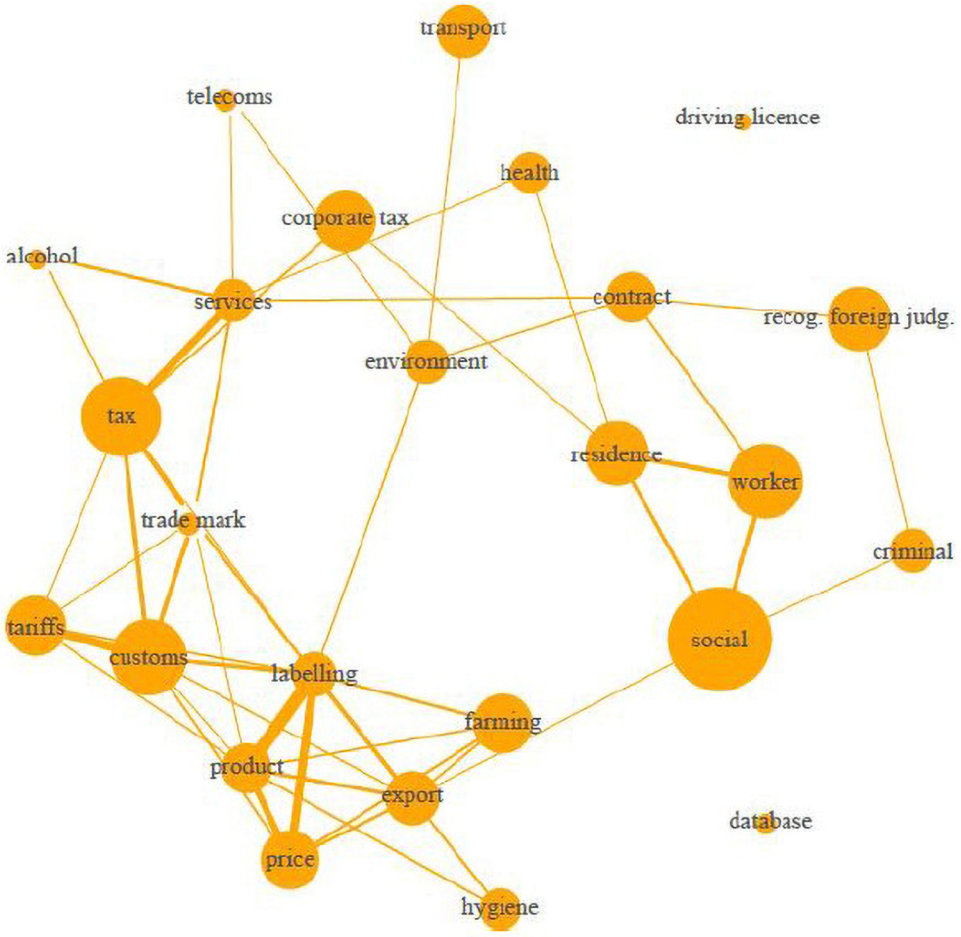
It is also possible to construct dynamic topic models to study the evolution of case law over time or 'litigant' topic models to study how issues vary across litigant types.

Recent work has applied topic modelling to study relative issue emphasis across infringement, annulment and preliminary rulings, highlighting how the CJEU's case law is influenced by the litigation agenda of case initiators (like the European Commission);[16] to compare topic salience in European Union legislation, CJEU rulings and contributions to the *Common Market Law Review*;[17] to explore Dutch Supreme Court decisions;[18] and to demonstrate the lingering centrality of market regulation in European Union law-making in the twenty-first century.[19] While significant efforts have been expended on manually classifying the legal areas addressed by US Supreme Court rulings, some authors have proposed topic modelling as a more efficient and more accurate alternative.[20] Work by Peter Grazl and Peter Murrel further illustrates how topic modelling can assist in exploring large collections of old legal texts. They apply topic modelling to reports of cases heard by English

16. A. Dyevre and N. Lampach, 'Issue Attention on International Courts: Evidence from the European Court of Justice', *Review of International Organizations* 1-23 (2020).
17. A. Dyevre, M. Ovadek and M. Glavina, 'The Voices of European Law: Legislators, Judges and Law Professors', forthcoming *German Law Journal* (2021).
18. Y. Remmits, *Finding the Topics of Case Law: Latent Dirichlet Allocation on Supreme Court Decisions* (2017).
19. N. Lampach, W. Wijtvliet and A. Dyevre, 'Merchant Hubs and Spatial Disparities in the Private Enforcement of International Trade Regimes', *International Review of Law and Economics* 105946 (2020).
20. D. Rice, 'Measuring the Issue Content of Supreme Court Opinions', 7 *Journal of Law and Courts* 107-27 (2019).

doi: 10.5553/ELR.000191 - ELR 2021 | No. 1

Figure 7 Topic model of CJEU preliminary rulings represented as network



between the fourteenth and eighteenth centuries (N = 52,949).[21]

4.4 Word Embeddings

Tools like LSA, PCA and topic modelling are typical of the BOW paradigm. Word embeddings, by contrast, are part of a new text-mining paradigm inspired by the defining principle of distributed linguistics – 'a word is defined by the company it keeps'.[22]

To explain how word embeddings work, the best is, again, to start with an example. Suppose you want to investigate variations in attention to a particular phenomenon, e.g. politics and populism in posts on a major legal blog. To measure attention to this concept, we might first try to come up with a list of keywords (e.g. 'politics', 'party', 'populism'…) capturing attention to this phenomenon and then determine the extent to which our keywords are actually matched in the document collection. However, this approach often delivers poor results because the same phenomenon can be characterised in many different ways, leading exact matches to either over- or underestimate the true number of relevant instances of attention to the phenomenon in question. (The frustrating feeling is surely one that many jurists have experienced when trying to retrieve documents via a keyword search in some legal database.)

Word embeddings help deal with this problem by representing words not as frequencies – the BOW approach – but as sequences (i.e. a vectors) of occurrence similarities, generated via a shallow neural network. For example, Table 1 displays the first 40 items in the vector of occurrence similarities yielded by a word embeddings model trained on the German-language contributions to the *Verfassungsblog* (a leading constitutional law blog) using the Word2Vec algorithm. The vector corresponds to the words *Politik* (politics), *Parteien* (parties) and *Populismus* (populism).[23]

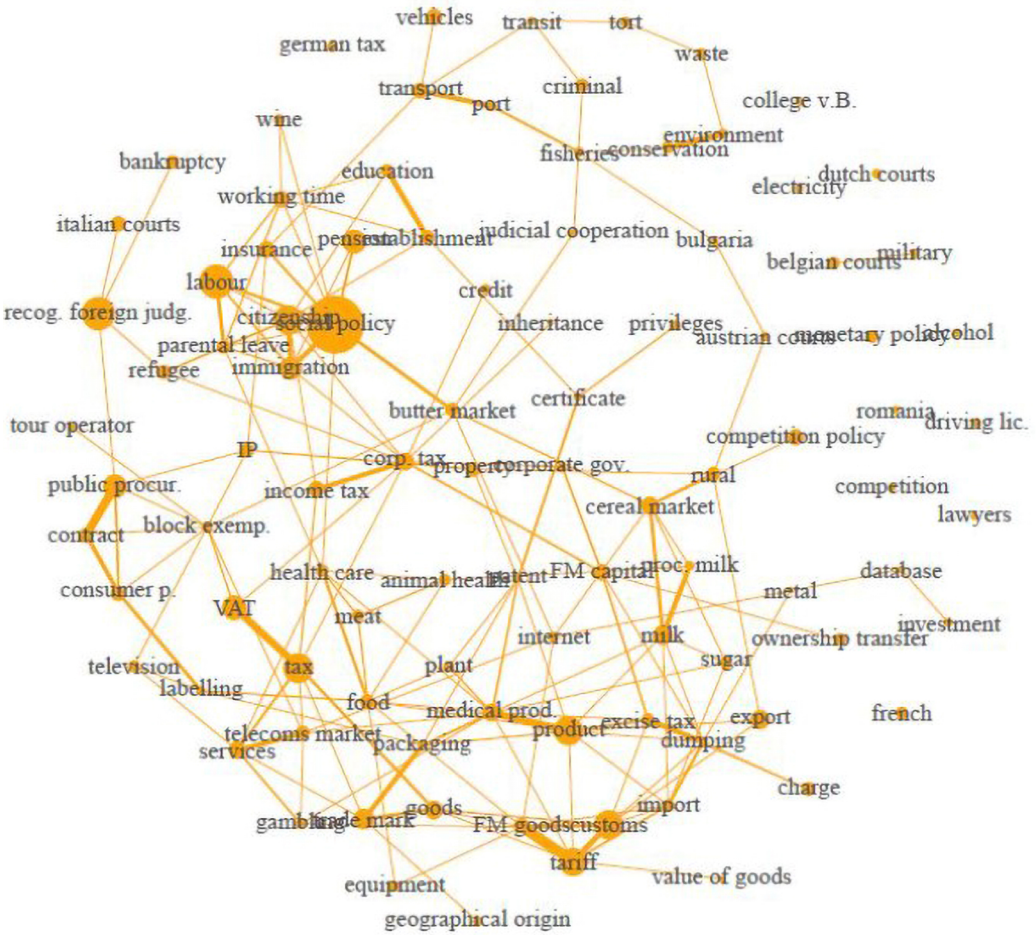
Numbers next to the words in Table 1 indicate the cosine occurrence similarity. The closer it is to 1, the more similar is the word's context of occurrence to that of *Politik* + *Parteien* + *Populismus*. Here the word exhibiting the highest cosine similarity score is *Eliten* (elites), which makes sense since elite-bashing is a defining feature of populist discourse. Other terms, including *Presse* (press), *Medien* (media) and *Bürger* (citizen), frequently

21. P. Grajzl and P. Murrell, 'A Machine-Learning History of English Case-law and Legal Ideas Prior to the Industrial Revolution I: Generating and Interpreting the Estimates', 17 *Journal of Institutional Economics* 1-19 (2021).

22. T. Mikolov et al., 'Distributed Representations of Words and Phrases and Their Compositionality', in C.J.C. Burges and L. Bottou and M. Welling and Z. Ghahramani and K.Q. Weinberger, *Advances in Neural Information Processing Systems* 3 (111-3119 (2013).

23. Several word embedding algorithms exist, including Word2Vec, Fasttext and Glove. Here we relied on the Word2Vec approach.

Figure 8 *Topic model of CJEU preliminary rulings with 100 topics*



come in populist rhetoric, too. *Verwaltung* (administration) and *Instrumente* (instruments), though, are less intuitively associated with politics, populism or partisan organisations.

Co-occurrence similarity here refers to the words that tend to appear around the target word. How many words before and after the target word should be considered – the window size – is one of the parameters that have to be set by the researcher before training an embedding model. A window size of 5 means that two words and two after the target word will be considered; a window size of 9 four words before and four words after; and so on. The neural network is then trained to predict either the target word from the surrounding words (continuous bag-of-words method) or the surrounding words given the target word (skip-gram method).

As with machine learning and neural networks in general, the more data (texts) the better. This is why instead of training embeddings from scratch on a relatively small collection of blog posts, it may be preferable to use a pre-trained model built on a much larger corpus. Table 2 shows the words associated with *Politik* + *Parteien* + *Populismus* from a pre-trained embeddings model constructed from all German Wikipedia pages, with a vocabulary of nearly five million words.[24] Pre-trained embeddings constructed from legal documents also exist.[25]

The cosine similarity scores are generally higher in Table 2 than in Table 1, which suggests that the pre-trained model better captures contextual similarity. In fact, it assigns a high cosine similarity scores to typos like 'poltk' (cosine = 0.846). This is because typos appear in the same context as the word with the correct spelling. The similarity scores assigned to typos highlight how word embedding models handle synonymy, which represents a major advance for legal information retrieval tasks.

That pre-trained embeddings can deliver better results than locally trained embeddings (i.e. embeddings trained on the corpus one actually wants to investigate) illustrates the notion of transfer learning. What a model learns about language use from a very large corpus is often transferable to smaller text collections.

24. A wide range of word embedding models spanning multiple languages can be downloaded from a repository made available by the Language Technology Group at the University of Oslo; *see*http://vectors.nlpl.eu/repository (last visited 12 November 2020).
25. I. Chalkidis and D. Kampas, 'Deep Learning in Law: Early Adaptation and Legal Word Embeddings Trained on Large Corpora', 27 *Artificial Intelligence and Law* 171-98 (2019).

doi: 10.5553/ELR.000191 - ELR 2021 | No. 1

Table 1 *Top 40 occurrence similarity scores for vector Politik + Parteien + Populismus from embeddings (Word2Vec) trained on German-language contributions to the Verfassungsblog*

eliten 0.8215773105621338	veränderungen 0.7256141901016235
vorteile 0.7712808847427368	gerechtigkeit 0.7244186401367188
öffentlichkeit 0.7557699084281921	medien 0.7238696813583374
bürger 0.7556987404823303	bevölkerung 0.7218047976493835
institutionen 0.7544448971748352	instrumente 0.7199758887290955
positionen 0.7507109642028809	kultur 0.7188452482223511
denjenigen 0.7492086291313171	verfassungen 0.7175022959709167
presse 0.7491012215614319	verteidiger 0.7158944606781006
vielfalt 0.7490779757499695	schmieden 0.7158849239349365
minderheiten 0.7462370991706848	verhältnisse 0.7151006460189819
nationalstaaten 0.7429373264312744	etablierten 0.7142930626869202
demokratien 0.7417148351669312	wissenschaft 0.7125871777534485
arena 0.7414146065711975	ideen 0.711867094039917
repräsentanten 0.7355824112892151	unionsbürger 0.7089967131614685
elite 0.7347079515457153	chancen 0.7073072791099548
gesellschaft 0.7327207922935486	debatten 0.7052429914474487
solidarität 0.7316428422927856	staatssekretäre 0.704701840877533
verwaltung 0.7315931916236877	justiz 0.7037882804870605
vernunft 0.7314342260360718	kommunikation 0.7020408511161804
wirtschaft 0.7265527248382568	minderheit 0.701974093914032

One powerful application of word embeddings is to generate weighted lexicons, which can be utilised to detect attention to a particular phenomenon or concept of interest. Figure 9 plots the variation in attention to politics, parties and populism in German-language contributions to the *Verfassungsblog* using the words contained in the vector *Politik* + *Parteien* + *Populismus* to measure the average attention to the underlying phenomenon.

Over time, the *Verfassungsblog* has been posting a growing number of English-language contributions. Figure 10 charts attention to the same phenomenon in English-language posts using the vector *politics* + *parties* + *populism* generated by Google's pre-trained word embedding model (Word2Vec) for English – which boasts a vocabulary of three billion words trained on Google News data.

These are potentially interesting results for scholars interested in the evolution of European constitutional law scholarship and a possible shift from a legalistic, narrowly doctrinal conception of legal scholarship to one that pays greater heed to political behaviours and social dynamics.[26]

To further illustrate the potential of word embeddings for attention detection and document retrieval, note that we can vary the specification of vectors to improve results or to capture conceptual nuances. The vector generated for *politics* alone will be different from the vector generated for *politics* + *parties* + *populism*. But if we wanted to generate a vector for terms associated with politics and political parties but not with populism, we could specify a vector like *politics* + *parties* – *populism*. Remarkably, in the Google pre-trained model, specifying *king* – *man* generates a vector in which the word with the highest cosine similarity score is *queen*.

So, by comparison with document search engines based on exact keyword matching, word embeddings provide a considerably more powerful tool to capture attention to concepts.

Another application of word embeddings is to compare change in the connotations of words across time, in

26. B. Caiepo and F. Benetti, 'How Political Turmoil is Changing European Constitutional Law: Evidence from the Verfassungsblog', *Verfassungsblog* (2020), https://verfassungsblog.de/how-political-turmoil-is-changing-european-constitutional-law-evidence-from-the-verfassungsblog/ (last visited 9 November 2020).

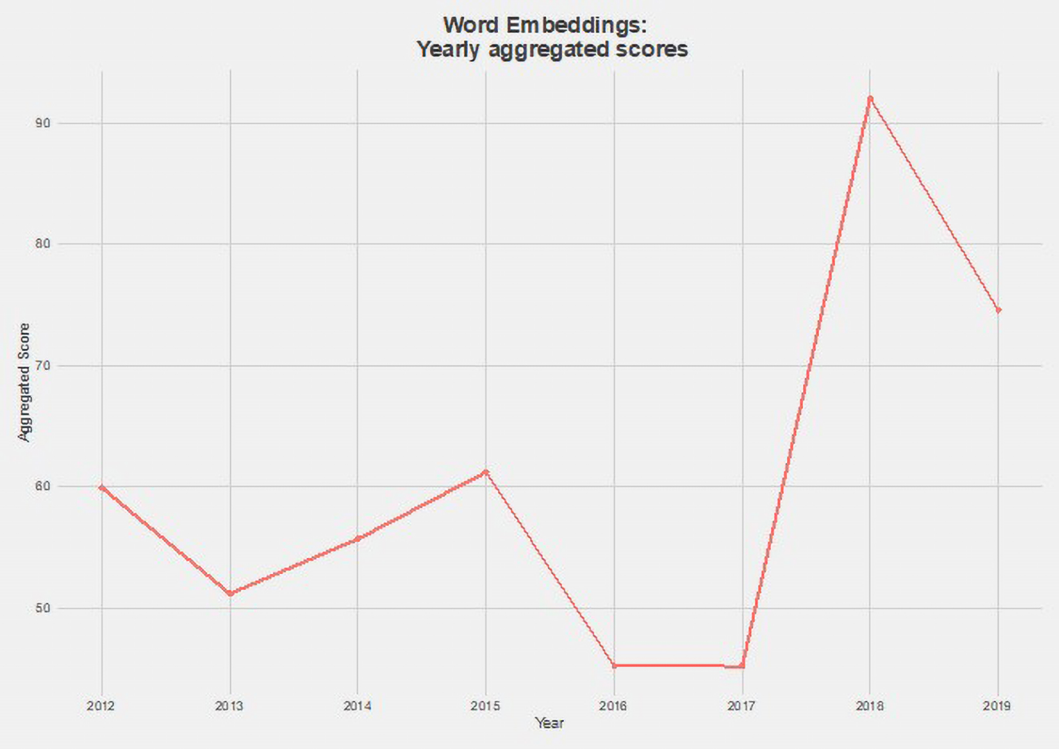
Table 2 *Top 40 occurrences for vector Politik + Parteien + Populismus from embeddings trained on German Wikipedia pages*

parteitaktik 0.8728734850883484	europapolitik 0.8352898359298706
parteipolitik 0.8678461909294128	hinterzimmerpolitik 0.8350450992584229
parteienpolitik 0.8615270853042603	demokratiedebatte 0.8343005180358887
parteienoligarchie 0.8604286313056946	demokratieverachtung 0.833306074142456
klientel- 0.8596892952919006	demokratieverlust 0.833281397819519
einheitsparteien 0.8587565422058105	eu-kritischer 0.8332792520523071
demokratie 0.8570675849914551	nationalpopulistischen 0.8332092761993408
populisten 0.8544521331787109	systemopposition 0.8329799771308899
euro-kritik 0.8523470163345337	linkspopulisten 0.832958459854126
klüngelei 0.8509451746940613	stimmungskanzlerin 0.8328656554222107
poltik 0.8463006615638733	schröder-ära 0.8327784538269043
stimmungsdemokratie 0.8435574173927307	politischen 0.8327664136886597
eu-zentralismus 0.8430708050727844	politikeliten 0.8319368362426758
parteienkartell 0.8392568826675415	europafeindlichkeit 0.8310469388961792
regierungspolitik 0.839039146900177	wirtschaftslobbyismus 0.8308700323104858
partikularinteressen 0.83860182762146	eurorettungspolitik 0.8304780125617981
parteiengezänk 0.8367708325386047	konzernspenden 0.8304095268249512
troika-politik 0.83570396900177	protestparteien 0.8301451206207275
linkspopulismus 0.8356888890266418	euroskepsis 0.830102801322937
wirtschaftslobbyisten 0.8353110551834106	parteitagsbeschlüsse 0.8300416469573975

Figure 9 *Relative incidence of words relating to 'Politik', 'Partei' and 'Populismus' in German-language contributions to the Verfassungsblog, 2009-2019*

doi: 10.5553/ELR.000191 - ELR 2021 | No. 1

Figure 10 *Relative incidence of words relating to 'politics', 'parties' and 'populism' in English-language contributions to the Verfassungsblog, 2012-2019*



which case separate embeddings models are trained on subsets of the corpus corresponding to distinct periods.[27] A noted study by Elliott Ash, Daniel Chen and Arianna Ornaghi has applied a similar approach to compare gender stereotypes across the opinions of federal judges in the United States.[28] 380,000 judicial opinions were grouped by authorship, and separate embedding models were trained for each judicial author. The distance between the vector *male – female* and *career – family* was then used to construct a gender slant indicator. The authors find that judges for whom these two vectors are closer – meaning that they more closely associate men and women with traditional gender roles – vote more conservatively on women's rights' issues such as reproductive rights, sexual harassment and gender discrimination. Moreover, they are less likely to assign opinions to female judges but are more likely to reverse lower-court decisions if the lower-court judge is a woman, and they cite fewer female-authored opinions.

A related study by Douglas Rice, Jesse Rhodes and Tatishe Nteta has examined racial biases in a corpus comprising over 1 million state and federal court opinions. The authors find stereotypically African-American names to be systematically associated with more negative words compared with stereotypically European-American names.[29]

4.5 Document Clustering with Word Embeddings: Doc2Vec

Closely related to the word embedding approach just described is a document clustering technique known as Doc2Vec. It relies on the same representation of words, but instead of training the neural network to predict only the target word or the surrounding terms, it is also trained to predict the documents in which they occur. Documents thus become associated with word vectors.

Doc2Vec is similar to PCA/LSA in that it simultaneously relates words and documents. The principal difference, however, is that Doc2Vec draws on a much more sophisticated word representation.

A Doc2Vec model can be visualised by means of a t-SNE (shorthand for 't-distributed stochastic neighbour embedding') plot. A t-SNE plot brings the high-dimensional vector representations of documents into a format where similarities among documents are easier to appreciate.

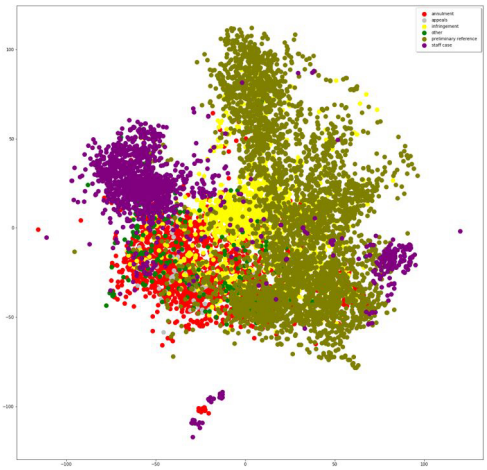
Figure 11 shows a t-SNE plot of a Doc2Vec model of European Court of Justice rulings, with colours denoting the procedure. The horizontal and vertical axes of a t-SNE plot are not amenable to substantive interpretation. But spatial proximity reflects similarity in word usage. Here the plot suggests some degree of overlap across procedures but greater heterogeneity in rulings originating in preliminary references.

27. Studies adopting this approach have revealed the evolution of gender and ethnic stereotypes or the changing connotations of the word 'gay'; *see* W.L. Hamilton, J. Leskovec and D. Jurafsky, 'Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change', ArXiv Prepr. ArXiv160509096 (2016); N. Garg et al., 'Word Embeddings Quantify 100 Years of Gender and Ethnic Stereotypes', 115 *Proceedings of the National Academy of Sciences* E3635-E3644 (2018).

28. E. Ash, D.L. Chen and A. Ornaghi, *Stereotypes in High-Stakes Decisions: Evidence from US Circuit Courts* (2020).

29. D. Rice, J.H. Rhodes and T. Nteta, 'Racial Bias in Legal Language', 6 *Research & Politics* (2019), doi: 10.1177/2053168019848930.

Figure 11 T-SNE plot of Doc2Vec model of European Court of Justice rulings (colour denotes procedure)



Looking at a large corpus of US Court of Appeals rulings, Daniel Chen and Elliott Ash have explored a variety of possible uses of Doc2Vec for the analysis of judicial opinions.[30]

Because precedents play an important role in legal argumentation, several studies have proposed Doc2Vec as a methodology to identify and measure case similarity.[31]

5 Supervised Classification Methods

Unsupervised approaches produce models and output without human input, which may seem to be a great advantage. However, the models and output generated by unsupervised methods always require ex post human interpretation. There is no absolute guarantee that the topics generated by a topic model will make sense or that the dimensions produced by an LSA model will be interpretable. This is not necessarily a problem if unsupervised techniques are primarily used for exploratory purposes. However, if one purports to rest an empirical assertion on the results of unsupervised methods, some human validation of at least a subset of these results may be required in order to demonstrate intersubjective validity.

Supervised methods, by contrast, do not require ex post validation because they seek to 'emulate' what humans do by discovering patterns in documents labelled by human annotators prior to training.

5.1 Obtaining Labelled Documents

Supervised approaches all require labelled documents. There are only two ways of obtaining labelled data. The first is to rely on documents that other researchers have already annotated. To measure the ideological direction of US federal court opinions, Carina Hausladen, Marcel Schubert and Elliott Ash were able to leverage an existing database (the Songer Database) where ideological direction had been hand-coded for a subset (5%) of federal cases. These annotated opinions were then used to train and test a range of algorithms.[32] Using labelled data sets from outside the legal domain can be tempting. But results may then have to be interpreted with caution. One study, for example, has sought to leverage academic articles from moral philosophy that had been labelled either as 'deontological' or 'consequentialist' to train machine learning classifiers to detect modes of moral reasoning in US federal opinions.[33] However, given the risk of low domain adaptation (the language of academic articles and judicial opinions may diverge too much), the results of studies adopting this strategy should be taken with a grain of salt.

When no labelled data set exists, the only way to obtain labelled data is to build it from scratch. In many areas, supervised machine learning projects rely on crowdsourcing platforms such as Amazon Mechanical Turk, where annotators recruited online tag documents for a modest compensation (and so at a low cost for the researcher).[34] However, crowdsourcing works best when a task is simple, quick and straightforward. So the specificity, technicality and complexity of legal language means that the crowdsourcing approach is not well suited to legal projects.

30. E. Ash and D.L. Chen, 'Case Vectors: Spatial Representations of the Law Using Document Embeddings', *Law as Data, Santa Fe Institute Press, ed. M. Livermore and D. Rockmore* (2019).
31. T. Novotná, 'Document Similarity of Czech Supreme Court Decisions', 14 *Masaryk University Journal of Law and Technology* 105-22 (2020); L.T.B. Ranera, G.A. Solano and N. Oco, 'Retrieval of Semantically Similar Philippine Supreme Court Case Decisions using Doc2Vec', in 2019 *International Symposium on Multimedia and Communication Technology (ISMAC)* 1-6 (2019); P. Bhattacharya et al., 'Methods for Computing Legal Document Similarity: A Comparative Study', ArXiv Prepr. ArXiv200412307 (2020).
32. C.I. Hausladen, M.H. Schubert and E. Ash, 'Text Classification of Ideological Direction in Judicial Opinions', 62 *International Review of Law and Economics* 105903 (2020).
33. N. Mainali et al., 'Automated Classification of Modes of Moral Reasoning in Judicial Decisions', in R. Whalen (ed.) *Computational Legal Studies*, 77–94 (2020).
34. C. Grady and M. Lease, 'Crowdsourcing Document Relevance Assessment with Mechanical Turk', in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk* 172-9 (2010).

doi: 10.5553/ELR.000191 - ELR 2021 | No. 1

Law students potentially offer a solution to the document annotation challenge. Without being accomplished legal experts, they tend to be more comfortable with legalistic language and better at parsing judicial prose or statutory provision. It is possible to integrate legal annotation tasks in tutorials and interactive classes. In fact, annotating legal documents can be viewed as an excellent exercise for students to practise and perfect legal analytic skills.[35]

At the KU Leuven Law School, EUTHORITY Project researchers recently conducted an annotation experiment with 52 third-year law students. Two hours per week, the annotation team assembled to annotate Belgian constitutional and Supreme Court (Court of Cassation + Council of State) rulings for explicit references to EU law. The rulings of Belgian top courts vary in both length (from a little more than thousand to several hundred thousand words) and complexity (from relatively simple asylum cases to more arcane regulatory issues). Eventually, the team was able to annotate 519 rulings, which is a reasonable number of documents but obviously pales in comparison with the millions of annotated pictures[36] computer scientists and engineers are able to tap into to train image-recognition algorithms.

Conducting annotation tasks with large groups of students demands a good work flow. Annotators must first be trained to recognise the concepts and information that the labels have been designed to capture. Documents must then be distributed to annotators and annotated documents collected. To produce high-quality annotations, it is recommended to have two or more annotators independently annotating the same document. Discrepancies then have to be identified to compute inter-annotator agreement metrics. To bolster quality, a reconciliation procedure can also be put in place. Software and online platforms have been developed to facilitate the completion of document annotation tasks by teams of annotators. The aforementioned annotation project conducted in Leuven, for example, relied on the cloud version of the TagTog[37] platform (which the project was allowed to use free of charge in exchange for making the labelled data public). Some software solutions, such as WebAnno,[38] are open source and can thus be used free of charge but require local server installation – which can be technically involved, unless technical support is provided.

5.2 Bag-of-Words Methods

As with unsupervised techniques, supervised techniques initially relied on the BOW paradigm. Supervised BOW methods involved the same data preparation steps, including converting texts into document-term matrix format. In a supervised set-up, the document matrix will look very similar, except that it will contain one or more additional columns for the labels produced by human annotators.

Before trying out some classification algorithms, the next stage will be to divide the data into train and test data. As its name suggests, the train set will serve to train many versions of the algorithm, whereas the test set will serve to measure their performance and select the best one. Dividing the data into train and test sets is called the 'holdout' procedure and is only one of many sampling procedures. When the number of annotated documents is small (less than 1,000), it is recommended to use some 'cross-validation' procedure. Cross-validation procedures begin by dividing the annotated documents into several folds (e.g. 10). One of the folds then serves as a test set while the algorithms are trained on the remaining folds. This process is then iterated with a different fold until every fold has served as a training set. Performance is evaluated by looking at the average across test folds. This way cross-validation ensures that as much data as possible is used for training.

When we say that the train set serves to train 'many versions' of an algorithm, we mean many combinations of words correlated with the labels. How many versions of the algorithm are fitted to the train data is for the researcher to decide in light of time and computational constraints (fitting a broader range of possible combinations obviously takes more time).

All these competing versions of the algorithm are then tested against the test data. The version that best predicts the human annotations in that set is then selected as the winner.

By way of illustration, we trained several algorithms to predict the labels 'EU law' and 'no EU law' in the aforementioned student-annotated corpus of Belgian high court rulings. Because this data set is relatively small (519 documents), we employed a cross-validation procedure. We then fitted thousands of versions of a handful of popular algorithms: logistic regression, support vector machine (SVM), random forest and sequential neural network.[39] While explaining the technical specifications of these algorithms is beyond the scope of the present article, Table 3 reports the performance of the 'best version' of each of these algorithms.

The metrics reported in Table 3 are the ones typically used in supervised text-mining classification tasks. Precision indicates the proportion of documents predicted to contain references to EU law that truly do so. On this metric, logistic regression and sequential neural network did best, achieving a precision of 95%. Recall measures the proportion of documents human annotators labelled as featuring EU law that the algorithm was able to

35. Conducting legal AI projects also helps bring greater awareness of the potential of new technologies for legal research and legal practice while contributing to the modernisation of legal education; *see* A. Dyevre, 'Fixing European Law Schools', 35 *European Review of Private Law* 151-168 (2017).
36. *See*www.image-net.org/ (last visited 9 November 2020).
37. *See*www.tagtog.net/ (last visited 3 March 2021).
38. https://webanno.github.io/webanno/ (last visited 4 March 2021).
39. For a concise explanation of these algorithms I refer the reader to S. Yıldırım, '11 Most Common Machine Learning Algorithms Explained in a Nutshell', *Medium* (2020), https://towardsdatascience.com/11-most-common-machine-learning-algorithms-explained-in-a-nutshell-cc6e98df93be (last visited 9 November 2020). For a survey from the perspective of econometrics *see* M. Gentzkow, B. Kelly and M. Taddy, 'Text as Data', 57 *Journal of Economic Literature* 535-74 (2019).

Table 3 Performance metrics of algorithms trained to predict the presence of references to EU law in Belgian high court rulings

	Precision	Recall	F1	MCC
Logistic regression	0.95	0.75	0.84	0.70
SVM	0.83	0.83	0.83	0.63
Random forest	0.91	0.88	0.89	0.77
Sequential neural net	0.95	0.75	0.84	0.70

retrieve. Here random forest did best, retrieving 88% of the documents thus labelled. F1 is a metric that combines precision and recall into a single number. The Matthews correlation coefficient (MCC) is yet another performance metric. It is recognised as the most reliable metric to evaluate a binary classifier because it takes into account the proportion of true negatives (documents predicted to feature no EU law and do not), false negatives (documents predicted to feature no EU law but that actually do), true positives (documents predicted to feature that really do) and false positives (documents predicted to feature EU law but that do not). Here random forest performs best with MCC = 0.77.

Similar BOW supervised approaches have variously been used to predict the outcome of ECHR cases;[40] the ideological direction of US federal opinions;[41] and to detect unfair clauses in online terms of service.[42]

5.3 Transfer Learning and Transformers

The new state of the art in supervised document classification draws its strength from several advances. The first is a revolutionary self-attention mechanism, known as 'transformer', which supports rich, contextualised representations of lexical and sentence meaning.[43] The second are new training methods. Models are trained to predict target words and whether two sentences appear next to each other. The third is greater leverage of transfer learning. Models are pre-trained, without human supervision, on vast repositories of texts. This knowledge can then be transferred to 'local' supervised tasks with additional fine-tuning steps.

These advances are embodied in BERT, the path-breaking natural language processing model developed by Google researchers.[44]

Based on a deep neural network architecture, BERT is able to focus attention on a given word in a sentence while simultaneously identifying the context of all the other words in relation to that word. The 'static', type-based, word embeddings discussed in the previous section represent a word as a vector of co-occurrences with cosine similarity scores reflecting co-occurrence frequencies. This permits static word embedding to handle synonymy (if, for instance, 'car' and 'vehicle' are used to mean the same thing they will have high cosine similarity score) but not polysemy or co-reference resolution (to determine what a pronoun refers to). The vector representing the word 'party', for instance, will not differentiate between party as in 'political party' and the party to a legal case. In a large and relatively diverse corpus, the vector is thus liable to assign high cosine similarity to words associated with both usages (e.g. 'political' and 'court'). By contrast, transformer models like BERT go beyond generalising across contexts. They represent words as dynamic, token-based vector embeddings, thereby coming much closer to capturing the particular, sentence-specific context of occurrence of a word. This, in turn, enables BERT to handle polysemy and co-reference resolution much better than previous language models.

The original BERT was trained on a giant corpus of GoogleBooks (800 million words) and Wikipedia pages (2.5 billion words) without human supervision by simply feeding it raw texts. Yet the power of BERT for supervised classification lies in the possibility to further fine-tune the pre-trained BERT on a 'local' data set. What has been learned from the giant corpus can thus be transferred to the local, smaller data set of direct interest to the researcher. Obviously, there are many linguistic patterns that no algorithm will be able to learn from a small data set. But a small data set may also instantiate specific patterns absent in the giant data set. In short, transfer learning helps combine the strengths of both data sets. Technically, local fine-tuning adds an additional layer of neurons to the neural network, thereby incorporating the local knowledge into the larger model.

BERT has been shown to outperform other algorithms on a wide range of natural language processing tasks.[45] One study has shown BERT to perform well at predicting the issue area codes of EU legislative acts.[46]

Table 4 reports the confusion matrix and performance metrics of a BERT model trained to predict whether EU legislative acts will be litigated. The data set was built by matching EU legislative acts in the EUR-Lex

40. M. Medvedeva, M. Vols and M. Wieling, 'Using Machine Learning to Predict Decisions of the European Court of Human Rights', 28 *Artificial Intelligence and Law* 237-66 (2020).
41. Hausladen, Schubert, and Ash, above n. 31.
42. Ranera, Solano, and Oco, above n. 30.
43. A. Vaswani et al., 'Attention is All You Need', in I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, *Advances in Neural Information Processing Systems* 5998-6008 (2017).
44. J. Devlin et al., 'Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding', ArXiv Prepr. ArXiv181004805 (2018).
45. *Id.*
46. I. Chalkidis et al., 'Large-scale Multi-label Text Classification on EU Legislation', ArXiv Prepr. ArXiv190602192 (2019).

Table 4 Confusion matrix and performance metrics of a BERT model trained to predict whether an EU legislative act will be litigated before the CJEU

		Predicted		
		Positive	**Negative**	**Total**
Actual	**Positive**	1,220	280	1,500
	Negative	664	4,836	5,500
Precision (positive): 0.6476				
Recall (positive): 0.1833				
F1 Score: 0.721				
Matthews correlation coefficient: 0.6408				

database to CJEU rulings. Only 3% of all EU legislative acts are ever litigated, and the probability that a given piece of legislation will be litigated in a particular case is very low.

When an outcome is a rare event, as with our example, it is important to think carefully about what the bar for a good model should be for the task at hand. Indeed, inexperienced lawyers and laypeople are often impressed by headline metrics like '90% accuracy'. However, achieving 90% correct classifications may, in many settings, be indicative of a poor performance. In fact, it all depends on the task and data set. With our EU litigation data set, it would have been easy to achieve 97% accuracy, since a model predicting that EU legislative acts are never litigated would be right 97% of the time. So here accuracy is a misleading metric, and precision and recall for the rare outcome provide a better gauge of performance.

Table 4 reports results for a subset of the data, where CJEU decisions featuring EU legislative acts have been deliberately oversampled. Oversampling the rare outcome is important to ensure that the algorithm has enough information to learn the patterns associated with this outcome.

While it is certainly possible to improve on these results through further local fine-tuning, a precision of 0.65 (i.e. out of 1,884 predicted to be litigated, 1,220 actually are) and a recall of 0.81 (i.e. out of 1,500 litigated, 1,220 were predicted to be so) are encouraging results.

Since the release of the first BERT, new variants of BERT have appeared, pre-trained on a wide range of general (RoBERTa) or domain-specific corpora (BioBERT, sciBERT…) in a variety of languages (e.g. robBert in Dutch, flauBERT in French, etc.). Multilingual BERT models, simultaneously pre-trained on multiple languages, have been shown to support transfer learning across languages. BERT models pre-trained on large collections of legal documents have also been released to assist with legal classification and prediction tasks.[47]

The arrival of BERT has triggered an AI race where research teams at big tech firms are vying to attain ever-higher performance with increasingly complex transformer language models: RoBERTa (Facebook), XLNET (Google), GPT-2 (OpenAI), Turing NLG (Microsoft) … The last such model to outperform its rivals, GPT-3 from OpenAI, boasts 175 billion parameters (by comparison, BERT has only 110 million parameters). The pace of technological development holds out great promise for the future of legal text-mining research and natural legal language understanding.

While transformers have just come along and applications to the legal domain are only starting to appear in publications and conference proceedings, an article by Evan Gretok, David Langerman and Wesley Oliver provides an interesting illustration of the application of transformer models to the study of legal doctrines. The authors trained transformer-based algorithms to classify rulings pertaining to the Fourth Amendment of the US Constitution depending on whether they applied a bright-line or a totality-of-the-circumstances rule. The best model (based on BERT) achieves an accuracy of 92%.[48]

As researchers begin to realise the potential of natural language processing for large-scale doctrinal analysis, we should expect to see many studies along these lines in the near future.

In the multilingual context of continental Europe, researchers may further seek to leverage the power of multilingual transformers to develop legal documents classifiers or predictors that can be deployed across multiple jurisdictions.

47. *Id.*

48. E. Gretok, D. Langerman and W.M. Oliver, 'Transformers for Classifying Fourth Amendment Elements and Factors Tests', *Legal Knowledge and Information Systems* JURIX 63-72 (2020).

6 Learning Text-Mining Methods

How can lawyers with no prior training in machine learning or data science get started?

One answer (at least for the motivated reader) is to learn a programming language like Python either by following one of the many free online tutorials or by taking a class at a nearby university campus. Python[49] is the language of choice for machine learning, text-mining and data harvesting tasks and the most popular among researchers and developers. Its ecosystem of libraries support the latest models and algorithms. While some lawyers may find the mention of 'programming' off-putting, Python is actually a very intuitive programming language. Moreover, the libraries provide many shortcuts that make it possible to complete a task with very few lines of code.

An alternative to Python is R, another popular programming language with many libraries designed for text-mining tasks, from data-harvesting to topic modelling and LSA/PCA. R was primarily developed for statistical analysis and does not support more advanced embeddings and transformer methods.

Both Python and R, along with their libraries, are entirely open source. They all can be downloaded and installed from the internet. The same goes for the pre-trained embeddings and transformers mentioned in this article (except for GPT-3).

Finally, for those who would prefer to avoid any kind of programming, RapidMiner comes with a graphical user interface to carry out end-to-end text-mining tasks without writing code.[50] Unlike Python and R, RapidMiner is a commercial platform. Yet its free version supports a wide range of supervised as well as unsupervised methods for data sets with up to 10,000 rows.

7 Conclusion

Text-mining and natural language understanding have been making great strides in recent years. Some of these techniques are at the heart of the hyper-hyped 'AI revolution' and are fuelling the development of legaltech.

To be sure, anyone who has actually attempted to use the techniques surveyed here will have realised that algorithms do not process language the way humans do. All techniques, even the most advanced ones, have limitations. Yet, thanks to their scalability, they open up a new possibility for legal research to explore and canvass vast repositories of legal documents.

There exist many variants of the techniques reviewed in this article and many more tasks to which they either already have been or may potentially be applied. However, I hope that the illustrations provided here and the techniques surveyed give the reader a sense of the potential that these techniques offer for academic legal research.

49. www.python.org (last visited 9 November 2020).
50. https://rapidminer.com (last visited 9 November 2020).