

The Case for a Linked Data Research Engine for Legal Scholars

Kody MOODLEY* , Pedro V HERNANDEZ-SERRANO**, Amrapali J ZAVERI**, Marcel GH SCHAPER***, Michel DUMONTIER** and Gijs VAN DIJCK***

This contribution explores the application of data science and artificial intelligence to legal research, more specifically an element that has not received much attention: the research infrastructure required to make such analysis possible. In recent years, EU law has become increasingly digitised and published in online databases such as EUR-Lex and HUDOC. However, the main barrier inhibiting legal scholars from analysing this information is lack of training in data analytics. Legal analytics software can mitigate this problem to an extent. However, current systems are dominated by the commercial sector. In addition, most systems focus on search of legal information but do not facilitate advanced visualisation and analytics. Finally, free to use systems that do provide such features are either too complex to use for general legal scholars, or are not rich enough in their analytics tools. In this paper, we motivate the case for building a software platform that addresses these limitations. Such software can provide a powerful platform for visualising and exploring connections and correlations in EU case law, helping to unravel the “DNA” behind EU legal systems. It will also serve to train researchers and students in schools and universities to analyse legal information using state-of-the-art methods in data science, without requiring technical proficiency in the underlying methods. We also suggest that the software should be powered by a data infrastructure and management paradigm following the seminal FAIR (Findable, Accessible, Interoperable and Reusable) principles.

I. INTRODUCTION

This article explores a software and data infrastructure that can inspire and answer new research questions in empirical legal research, that could not be answered previously with traditional human analysis of legal information. In particular, it will look at how this infrastructure can be created to prepare data automatically, and apply data analysis methods, such as Natural Language Processing (NLP)¹ and network

* Faculty of Law, Maastricht University; email: kody.moodley@maastrichtuniversity.nl.

** Institute of Data Science, Maastricht University

*** Faculty of Law, Maastricht University.

¹ CD Manning and H Schütze, *Foundations of statistical natural language processing* (MIT Press 1999); G Boella et al, “Semantic relation extraction from legislative text using generalized syntactic dependencies and support vector machines” in *International Workshop on Rules and Rule Markup Languages for the Semantic Web* (Springer 2013)

analysis,² to case law on behalf of the legal scholar, unburdening them from the need for technical expertise. The vision for this software is for use in legal academia as both an educational and research tool, building on similar systems in the literature that are limited either by their commercial nature, narrow functionality, or complex interfaces unsuitable for anyone who is not a data science expert.

This contribution is structured as follows. First, an overview will be provided of possible research questions that may be answered by applying advanced data science to case law (Section II). Second, we will discuss the strengths and limitations of existing software platforms that aim to help legal scholars find and analyse legal information (Section III). Third, we will discuss the functional and architectural requirements of software that can address the limitations of existing systems and provide legal researchers with tools for studying the law in an intuitive, user-friendly way (Section IV). Finally, we will summarise our contribution and discuss the feasibility of developing the proposed system, potential challenges with this and current technologies which can help to overcome them (Section V).

II. POTENTIAL OF TECHNOLOGY FOR LEGAL RESEARCH

Case law traditionally relies on human analysis, that is analysis without software or other technical aid. Legal researchers manually search, read, and interpret court decisions. In this process, technological assistance is commonly available in the form of online search facilities (keyword search in electronic case law databases).

Case synthesis is the method commonly applied by legal researchers and law students when analysing court decisions.³ This method essentially entails that case outcomes are compared with the facts of the cases, with the purpose of explaining the differences in outcomes by the differences in facts.⁴ As a result of the high cognitive workload involved in this type of reasoning, case law is commonly analysed based on a relatively small number of cases, at least compared to the whole body of case law that is available.

The consequence of how case law is commonly studied is that not all knowledge is utilised. Data science methods enable computers to consider vastly more cases than human scholars,⁵ and therefore offer the possibility to further unravel the law and how it works. Scholars who have become active in this domain refer to this

pp 218–225; R Nanda et al, “Concept Recognition in European and National Law” in *Proceedings of the International Conference on Legal Knowledge and Information Systems (JURIX)* (2017) pp 193–198.

² Eg D van Kuppevelt and G van Dijck, “Answering Legal Research Questions About Dutch Case Law with Network Analysis and Visualization” in A Wyner and G Casini (eds), *Legal Knowledge and Information Systems* (IOS Press 2017); MGH Schaper, “A computational legal analysis of Acte Clair Rules of EU law in the field of direct taxes” (2014) 6 *World Tax Journal* 77.

³ F Nievelstein et al, “The worked example and expertise reversal effect in less structured tasks: Learning to reason about legal cases” (2013) 38 *Contemporary Educational Psychology* 118.

⁴ JK Gionfriddo, “Thinking Like a Lawyer: The Heuristics of Case Synthesis” (2007) 40 *Texas Tech Law Review* 1.

⁵ E Ruppert et al, “LawStats – Large-Scale German Court Decision Evaluation Using Web Service Classifiers” in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction* (Springer 2018) pp 212–222.

perspective as “harnessing legal complexity” and “legal DNA”.⁶ The interconnectedness of institutions (eg legislatures, agencies, and courts), norms (eg due process, equality, and fairness); actors (eg legislators, bureaucrats, and judges), and instruments (eg regulations, injunctions, and taxes) through processes (eg trials, negotiations, and rulemakings) with feedback mechanisms (eg appeals to higher courts and judicial review of legislation) illustrate the legal complexity, and are reinforced by their embedding in network architectures (eg cross-references between statute provisions and judicial opinions, as well as hierarchies of intra-state, state, and local governance institutions) that frequently produce self-organising properties (eg doctrines or codified statutory law). Actors and users (actual and potential) of this system typically exercise bounded rationality, have only partial information, and are able to exercise varying degrees of control on overall system behaviour. Consequently, the legal system is a complex and constantly changing system of hundreds of thousands of interrelated legal documents.⁷

With respect to case law, questions can be raised regarding who (or what) interacts with who (or what), how the interactions changes over time, where information originates (eg how many court decisions per year, per field etc), how it flows and at what speed, and when, where and which legal arguments were used, and relationships between texts or actors.⁸ More specifically, various research questions may be raised, including (but not limited to):

- What are the cases surrounding the landmark cases?
- Which clusters of decisions can be distinguished?
- Are there other landmark cases that have remained undetected in the literature?
- How often and in which instances do national courts cite European case law?
- Do national courts directly cite, for example, European case law, or indirectly (eg a national court that cites the Supreme Court of that nation that cites European case law)?
- What legal arguments are constructed?
- Have certain legal topics or legal concepts gained or lost importance over time (eg since the introduction of new EU member states, after the introduction of new legislation)?
- How does the information (eg citations) flow within jurisdictions (eg within EU law, French law) and across jurisdictions (eg from EU law to German law)?
- Is the law from some countries, case law in particular, more influential in European case law compared to case law from other countries?
- Is case importance related to characteristics such as the country of origin or the Advocate-General?
- How does case importance change over time?

⁶ JB Ruhl et al, “Harnessing legal complexity” (2017) 355 *Science* 1377. DOI: 10.1126/science.aag3013.

⁷ *ibid.*

⁸ G van Dijk, *MoneyLaw (and Beyond)* (Eleven International Publishing 2017).

Answering questions such as those presented above requires a variety of analytical methods, ranging from network analysis, statistical methods to NLP, in order to identify certain combinations of words or even arguments. Depending on the selection of the method, one can perform either simpler or deeper analysis of legal information to generate insight. Current research has started to answer some of the aforementioned questions. For example, network analysis studies on 26,681 majority opinions by the US Supreme Court and the cases that cite them from 1791 to 2005 have exposed interesting patterns.⁹ In those studies, it has been found, among other things, that reversed cases tend to be more important than other decisions, that cases that overrule the reversed cases “quickly become and remain even more important” and that the Supreme Court carefully embeds overruling decisions in past precedent.

As an example of the mechanism of network analysis through which we can answer some of the research questions mentioned above, let us consider the question: “Are there other landmark cases that have remained undetected in the literature?”. A priori, we can make a note of all the landmark cases (as qualitatively accepted by legal scholars) concerning a particular legal topic. Thereafter, we can plot the case citation network of the decisions in the same legal topic. We can use node centrality measures¹⁰ such as *in-degree*, *betweenness*, *closeness* and *page rank* to rank the computational centrality of all the nodes (representing cases) in the network. We can then observe whether higher computational centrality correlates with the qualitative assessment of importance by legal scholars. We may then notice a pattern such as “all qualitatively identified landmark cases score highly on page rank and closeness, but vary widely on other measures”. Then, we may identify obscure cases which also scored highly in page rank and closeness but were not previously acknowledged as landmark, which could spark questions about what legal scholars define to be a landmark case.

Various other network analysis studies have emerged attempting to conduct these kinds of investigations,¹¹ but the number of network analysis studies applied to case law is disproportionate to the number of legal publications where doctrinal analysis prevails.

There are at least three important reasons why such research has not yet taken off. Firstly, proper data infrastructures are not readily available to automatically find, extract and prepare legal information for analysis by software. The data collection and preparation for the network analysis studies¹² previously mentioned involved significant manual labour. Technologies such as *web scraping*,¹³ in principle, allow this information to be automatically extracted in larger volumes from public case law websites, keeping in mind the associated legal issues.¹⁴ However, these technologies have not been integrated into software that legal scholars can easily use to retrieve

⁹ JH Fowler and S Jeon, “The authority of Supreme Court precedent” (2008) 30 *Social Network* 16; JH Fowler et al, “Network Analysis and the Law: Measuring the Legal Importance of Precedents at the U.S. Supreme Court” (2007) 15 *Political Analysis* 324.

¹⁰ NE Friedkin, “Theoretical foundations for centrality measures” (1991) 96 *The American Journal of Sociology* 1478.

¹¹ van Dijck, *supra*, note 8.

¹² Fowler and Jeon, *supra*, note 9; Fowler et al, *supra*, note 9.

¹³ R Mitchell, *Web Scraping with Python: Collecting More Data from the Modern Web* (O’Reilly Media 2018).

¹⁴ Z Gold and M Latonero, “Robots Welcome: Ethical and Legal Considerations for Web Crawling and Scraping” (2017) 13 *Wash JL Tech & Arts* 275.

and prepare the information they need for analysis. Secondly, existing software platforms that provide access to legal information generally only focus on searching and browsing,¹⁵ neglecting to include functions for analysing the information using cutting-edge data science methods. Thirdly, those systems that do provide some features for analysing, exploring and visualising patterns in the information, some of which will be introduced in the next section, do not provide interfaces that are user-friendly for non-data science experts in the legal domain.

III. COMPUTATIONAL TOOLS ADDRESSING LEGAL RESEARCH

Available tools that apply computational analysis to judicial texts, can be free to use for the public, or not (eg those with a commercial focus). Users of both categories can range from legal professionals in firms, to scholars and researchers in universities, to the general public. Below, we provide a (non-exhaustive) overview of some prominent existing tools that are applied or could be applied to judicial datasets, and we discuss their functional limitations.

Data analytics has been a profitable enterprise for many corporate organisations in the last decade. A substantial proportion of them also offer specialised services for analysing legal information. The result is that, by far, available software for supporting analysis (not purely search) of legal information is predominantly developed by commercial enterprises. Table 1 lists and characterises some prominent examples, some of which are elaborated on below. Our criteria for selecting the analytics tools to survey are: (1) the tools should have a graphical user interface (because the goal is to enable legal scholars with no programming or technical experience to use the platform); (2) the tool should be capable of analysing case law texts specifically; (3) the user interface should provide a visual way for representing results from its analysis. In the “Technologies used” column of Table 1, “Corpus analysis” refers to analysis of collections of legal documents, whether these are legal contracts, legislative texts, or court decisions.

ROSS: ROSS is a software research engine that uses artificial intelligence to semi-automate legal research, claiming to make it more efficient and less expensive.¹⁶ Its data sources include a comprehensive body of case law texts originating in the United States Supreme Court, Circuit Courts of Appeals, District Courts, Bankruptcy Courts, State Supreme and Appellate courts. These are also enriched with information from various federal speciality courts and a selection of administrative boards.

To use ROSS, the user types in a natural language question (eg “What is the standard for gross negligence in New York after 2004?”) and submits it to the system. ROSS then uses NLP to “understand” or interpret the question using its proprietary algorithms. The jurisdiction and time range, for example, are identified (“New York” and “after 2004”). Thereafter, it searches the body of case law using the identified information to find a list of passages in the text that are relevant to the question. It also looks at citation graphs of

¹⁵ R Winkels, “The openlaws project: Big open legal data” in *Proceedings of the International Legal Informatics Symposium* (IRIS 2015) pp 189–196.

¹⁶ <www.forbes.com/profile/ross-intelligence>.

Table 1: A summary of some prominent software platforms for performing semi-automated and automated analysis of case law and legislative texts

Organization	Legal Analytics Software	Free Version	Commercial Version	Source code	Data Sources Coverage	Technologies Used
EUCaseNet OpenLab	EUCaseNet	Y	N	Closed	EU	Network analysis Case citations analysis Descriptive statistics of case law Document search engine
Openlaws Gmbh	Open Laws	Y	Y	Closed	EU	Descriptive statistics of case law & legislation
UM/NLeSC	Dutch Case Law Analytics	Y	N	Open	The Netherlands Austria, Bulgaria, France, Germany, Italy and UK	Network analysis Descriptive statistics of case law RDF Linked data Semantic Web
EUCASES	ConsumerCases	Y	N	Closed	Luxembourg	Natural Language Processing
Le Gouvernement Luxembourgeois	Legilux	Y	N	Closed		Document search engine Network analysis RDF Linked data Document search engine Corpus analysis
ROSS	ROSS	N	Y	Closed	U.S.	Case citations analysis Machine learning Natural Language Processing
LexPredict	ContraxSuite LexSemble	Y N	Y Y	Open Closed	U.S. U.S.	Corpus analysis Descriptive statistics of firm performance Descriptive statistics of case law

Table 1: (Continued)

Organization	Legal Analytics Software	Free Version	Commercial Version	Source code	Data Sources Coverage	Technologies Used
LexMachina	Analytics Platform	N	Y	Closed	U.S.	Corpus analysis Machine Learning
	Analytics Apps	N	Y	Closed	U.S.	Natural Language Processing Document management Descriptive statistics of firm performance Search visualization
Ravel Law	RAVEL	Y	Y	Closed	U.S.	Case analytics Corpus analysis Descriptive statistics of case law
Thomson Reuters	Westlaw Edge	N	Y	Closed	U.S.	Document search engine Descriptive statistics of firm performance Document management
Docket Alarm	Docket Alarm	Y	Y	Closed	U.S.	Document search engine Descriptive statistics of firm performance
Judicata	Clerk	Y	Y	Closed	California	Document search engine Case analytics
Intraspexion	Intraspexion	N	Y	Closed	U.S.	Deep learning for text classification

the cases to identify other relevant case passages to “read”. Once the final list of texts is retrieved, the texts are ordered according to relevance using a combination of machine learning¹⁷ algorithms for analysing the grammatical structure of text and other techniques. ROSS is a commercial platform with no free version available. Thus it is not possible to examine the full functionality of the system without purchasing a licence.

LexPredict: LexPredict is a company that provides software products and advisory services for quantitative legal research. The principles underpinning the LexPredict platform were first developed at the Center for the Study of Complex Systems at the University of Michigan. LexPredict’s main clients are US law firms and corporate legal departments. LexPredict’s software is provided through a wide variety of systems. The data sources used by these systems are equally diverse; however, they all concern case law and legislation originating from the US.¹⁸ The LexPredict platform splits its functionality across multiple commercially licensed software applications including: LexSemble, CounselTracker, and LexReserve, as well as many data products, including both cloud-based application programming interfaces (APIs) and downloadable on-premise solutions, such as: contract database, tender offer database, regulatory and legal action database, etc.

LexPredict also develops publicly available open-source software such as LexNLP¹⁹ and ContraxSuite.²⁰ LexNLP focuses on recognising specific types of information from legal text. Some of these categories include: dates (eg effective and termination dates of contracts), parties (eg persons and organisations), citations of legislation or case law (eg “26 USC 501”), references to courts (eg “Supreme Court of New York”) and copyrights or trademarks (eg “(C) Copyright 2000 Acme”). ContraxSuite is the most similar product from LexPredict to the proposed software in this paper. It is a tool to analyse text in legal documents and provides dashboards and visual plots about patterns it identifies in legal texts. It can, for example, visualise clusters of similar legal documents in a graph (using algorithms to measure the prevalence of common and thematically similar terms in the documents). Figure 1 below displays one of the dashboards for this functionality.

Essentially, the platform builds upon the functionality of LexNLP by focusing on retrieval of relevant legal documents, identification of key clauses in the documents, and generation of reports with data-driven descriptions of the documents and relations between them. While the source code for the technologies underlying ContraxSuite is made publicly available for use by software developers, the source code for the complete software platform itself (with its graphical user interface) is not publicly available.

OpenLaws: EUR-Lex²¹ provides searchable access to legislation on the EU-level and case law texts from the European Court of Justice. OpenLaws²² is a software platform

¹⁷ AL Samuel, “Some Studies in Machine Learning Using the Game of Checkers” (1959) 3 IBM Journal 535.

¹⁸ DM Katz et al, “A general approach for predicting the behavior of the Supreme Court of the United States” (2017) 12 *PLoS ONE* e0174698. DOI: <[journals.plos.org/plosone/article?id=10.1371/journal.pone.0174698](https://doi.org/10.1371/journal.pone.0174698)>.

¹⁹ <github.com/LexPredict/lexpredict-lexnlp>.

²⁰ <github.com/LexPredict/lexpredict-contraxsuite>.

²¹ <eur-lex.europa.eu>.

²² Winkels, *supra*, note 15.

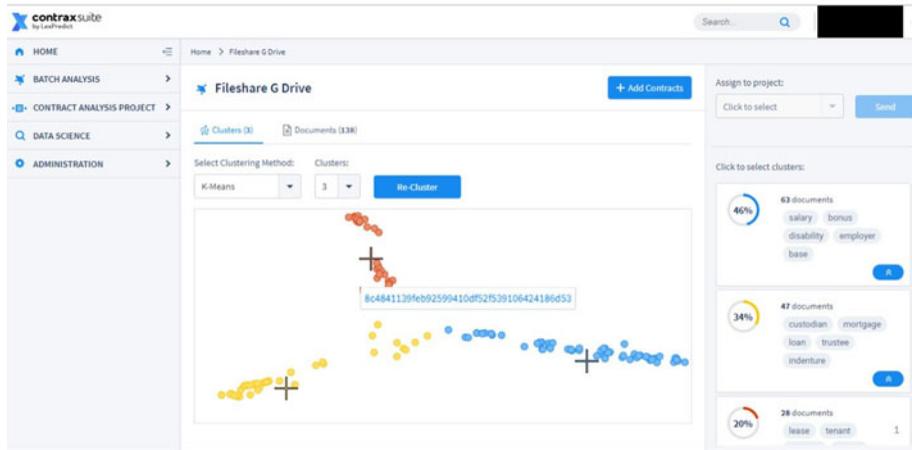


Figure 1: A dashboard within ContraxSuite for displaying relations between legal documents based on their semantic content

(website) that provides similar search and retrieval functions to EUR-Lex, with the added benefit of having a more intuitive and user-friendly search interface. It is available both as a free search tool and also through paid subscription, the latter providing users with an account to store and bookmark their searches and retrieved documents, as well as link them with other decisions if required. The aim of the tool is to facilitate the user's automatic retrieval of relevant legal documents. The task of reading, interpreting and analysing the content is still left up to the user. OpenLaws is based upon data extracted from EUR-Lex and Rechtsinformationsystem des bundes (RIS)²³ from Austria. A limitation of the OpenLaws platform is that it is mainly for search of relevant legislation and case law. It does not, for example, provide tools to perform network analysis or generate graphs to show relationships between cases and legislation. A helpful feature of the system is that one can create an account and store searches of documents that can be shared with others.

ConsumerCases: the ConsumerCases²⁴ software platform was generated from the EUCases²⁵ project which was a large and pioneering effort to provide EU case law and legislation in the Linked Data²⁶ format (also called Resource Description Format or RDF²⁷) that is recommended by the World Wide Web Consortium (W3C, the main international standards organisation for the World Wide Web). Linked Data is a paradigm espoused by Tim Berners-Lee, the inventor of the World Wide Web, and it inspired the creation of a data format to be used on the Web that represents information in such a way that it can be easily and semi-automatically linked to related information

²³ <www.ris.bka.gv.at>.

²⁴ More information about the ConsumerCases platform can be found at <eucases.eu> and the interested reader can contact info@empirica.com to gain access to the platform.

²⁵ <eucases.eu>; G Boella et al, "Linking legal open data: breaking the accessibility and language barrier in european legislation and case law" in *Proceedings of the 15th International Conference on Artificial Intelligence and Law* (ACM 2015) pp 171–175.

²⁶ T Berners-Lee, "Linked data – design issues" (2016) <www.w3.org/DesignIssues/LinkedData.html>.

²⁷ <www.w3.org/TR/rdf11-primer>.

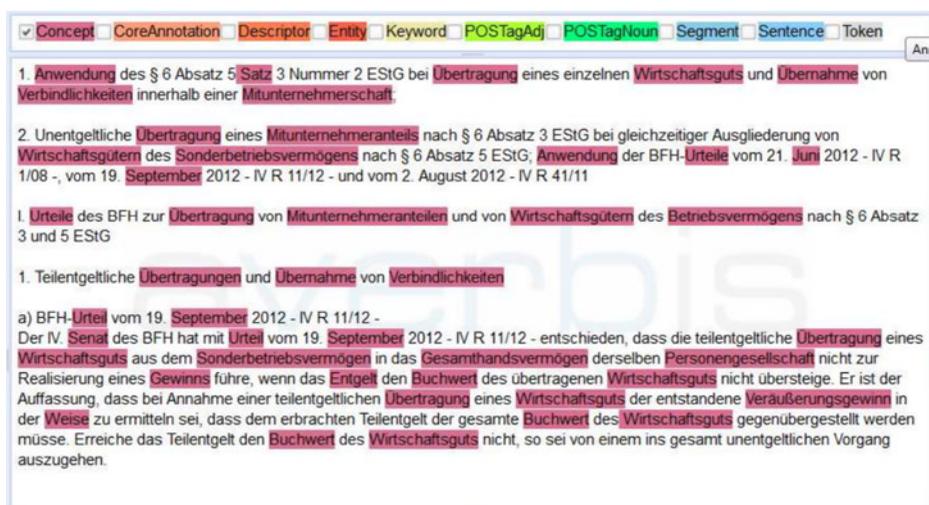


Figure 2: ConsumerCases feature for annotating legal texts with part of speech fragments, dates, organisations, and other relevant entities

so that more research questions can be answered (an example is illustrated in Section IV). The ConsumerCases platform is software that allows search and retrieval of legal documents, and beyond this, automatic annotation of the text with relevant entities (using NLP). This feature extends the power of a tool such as OpenLaws, which is purely focused on search. It allows the user to gain insight into the content of the text without having to read the entire text manually. With the click of a checkbox one can see the main entities (eg dates, legal persons, organisations, courts, articles cited etc) highlighted in the text (see Figure 2). This can save time in comprehending the content in the case. However, the task of identifying relationships or connections between cases is still left up to the user. In other words, graphical and visual tools to map cases and the citations between them using nodes and edges (network analysis) is currently missing. The ConsumerCases platform, while being publicly accessible to use online, requires login credentials that must be requested via email.

Maastricht University (UM) / Netherlands eScience Center (NLeSC) case law analytics: A case law analytics application was developed to analyse Dutch case law (see Figure 3).²⁸ The data was imported from *Rechtspraak.nl*²⁹ and the tool helps the user perform network (citation) analysis on the cases. The nodes in the graph represent cases and the edges represent citations between the cases. Various filter options are provided, including the selection of important decisions. The “importance” of individual court decisions are measured using standard graph or network analysis “centrality” measures³⁰ such as in-degree, out-degree, betweenness. Algorithms are also used to cluster related decisions to identify citation communities.

²⁸ <nlesc.github.io/case-law-app>.

²⁹ <www.rechtspraak.nl>.

³⁰ NE Friedkin, “Theoretical foundations for centrality measures” (1991) 96 *The American Journal of Sociology* 1478.

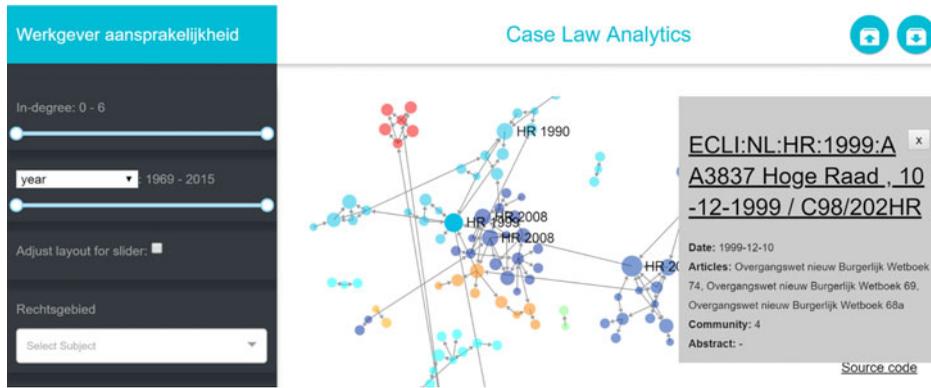


Figure 3: User interface of UM/NLeSC case law software to perform network analysis on Dutch case law from Rechtspraak.nl.

A useful feature of the UM/NLeSC tool is that it also provides information describing relevant properties (*metadata*) about each node (case). Databases like Rechtspraak.nl and EUR-Lex provide such metadata for cases (eg the judge, applicant, defendant, and lodge date). UM/NLeSC provides filters for the citation graph to keep only those cases that have certain properties (eg those that were decided within a certain data range). However, network analysis is inherently citation focused, which means that it looks purely at the citation behaviour between cases. Although citation analysis can be helpful in revealing landmark cases and the factors leading to a legal precedent, computer algorithms that analyse the content (full text) of the decisions can be helpful for detecting further connections between cases. Tools such as ConsumerCases allow users to automatically identify such information in the case texts but it would give legal scholars greater insight if the information could be attached as extra metadata to the nodes in the network analysis graph and exploited visually. Unfortunately, such information is missing from network analysis tools such as UM/NLeSC. Furthermore, UM/NLeSC is currently limited to Dutch case law which narrows its applicability for answering broader legal research questions. A tool which could build on the featureset of UM/NLeSC by integrating case law from national courts across the EU, linking these with decisions from the Court of Justice of the EU (EUR-Lex), and enabling network analysis on the data, would provide a valuable window into the flow of decisions between the national and EU-level.

EUCaseNet: EUCaseNet³¹ (see Figure 4) is a web-based software platform to perform analysis on EU case law specifically. It was developed by Lettieri et al³² and is based on case law published in the EUR-Lex database. The tool facilitates network analysis on the full body of case law from the Court of Justice of the EU. It has tools to perform network analysis using node centrality measures such as *betweenness*, *closeness* and *page rank*, and tools to perform descriptive statistics on EU case law focusing mainly on monitoring

³¹ N Lettieri et al, “Network, Visualization, Analytics. A Tool Allowing Legal Scholars to Experimentally Investigate EU Case Law” in *AI Approaches to the Complexity of Legal Systems* (Springer 2015) pp 543–555.

³² N Lettieri et al, “A computational approach for the experimental study of EU case law: analysis and implementation” (2016) 56 *Social Network Analysis and Mining* 6, no 1.

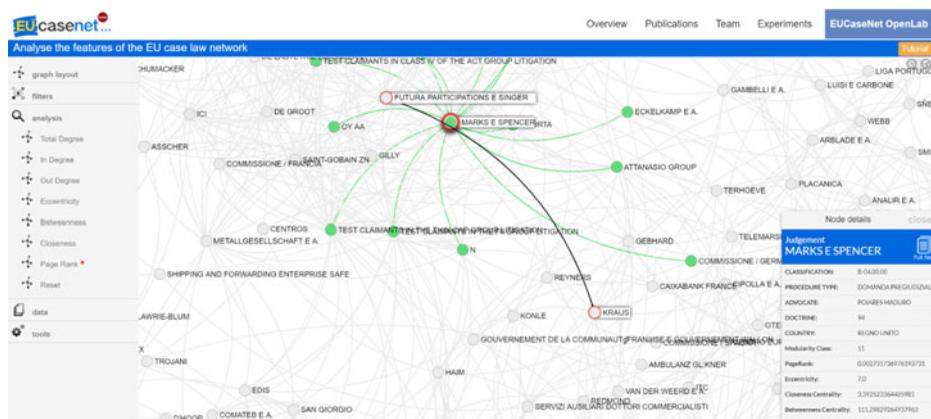


Figure 4: EUCaseNet dashboard for analysing EU case law using network analysis

what subject matters and topics the decisions tend to focus on and how this evolves over time. A dashboard is also provided to count the number of sentences (arguments) in case law transcripts that fall under each subject or topic of case law (eg EUCaseNet tags 1727 sentences in the body of EU case law as concerning topics related to agriculture and fisheries). Additionally, there is a feature which explores how computational measures of case importance or influentialness compare to the traditional perceptions of decision importance accepted through consensus by legal scholars. Of all the tools presented in this section, EUCaseNet is a powerful and useful research tool which most closely resembles the vision encompassed by the platform we present in this paper. However, there are some limitations which would warrant a larger feature-set. Firstly, the platform currently focuses only on EU case law. It does not attempt to connect these decisions to requests made from the national courts. It also does not facilitate computationally assisted analyses of case law texts using techniques such as NLP (such as that of ConsumerCases). Though it does provide descriptive statistics about the topics covered in EU case law and how this changes over the years, it does not provide other statistical information about cases (eg their duration with respect to topics and how this evolves over time, most cited articles etc).

Discussion: in this section, we have given an overview of some main software platforms that enable advanced analysis of judicial data in the EU and the US. Our main findings are that while the majority of the current platforms are dominated by the commercial sector, other platforms focus almost exclusively on search and retrieval of legal documents, and most platforms focus on case law from a single national database. Most of the platforms we surveyed are focused on case law from the US. In the EU, there are far fewer options for case law analytics software. Additionally, we could only identify one platform with a user interface (UM/NLeSC) which has made its software code completely public. The only non-commercial system that links case law from multiple EU member states, and goes beyond search and retrieval to do advanced analytics of case law, is ConsumerCases. While this platform provides NLP annotation of relevant entities in case texts, we could not retrieve this information to network analysis graphs and other visual aids that analyse

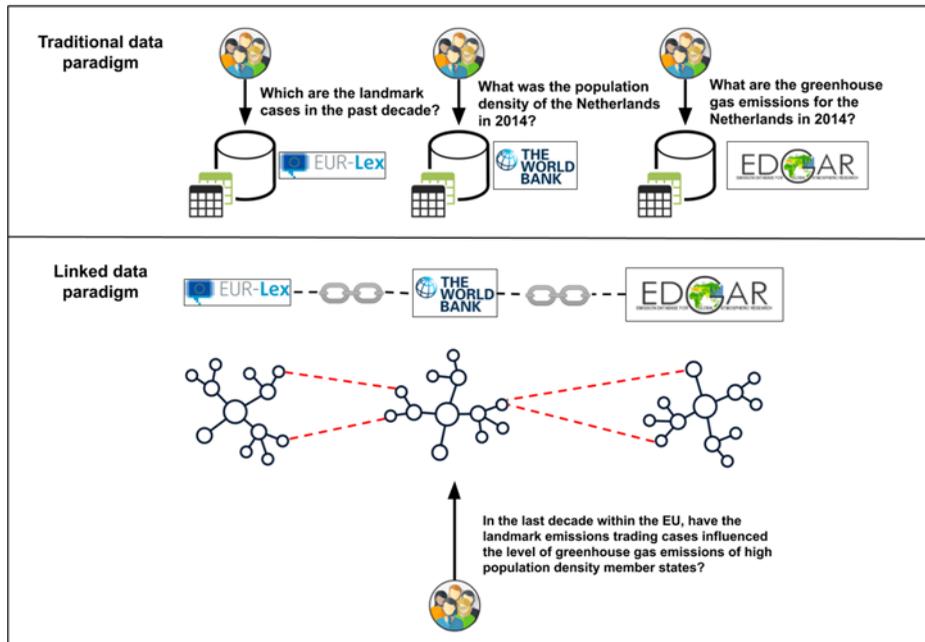


Figure 5: The Linked Data paradigm allows answering of research questions via single queries posed to a “knowledge graph”, as opposed to manual aggregation of answers across multiple queries and separate databases

case law over time, in order to make research easier for the average legal scholar. The only non-commercial system which analyses the full body of EU case law (EUR-Lex) is EUCaseNet. This platform has some functional limitations which have been discussed. However, it also has a potential drawback in how it represents the data it uses to power its analyses. That is, it does not take advantage of semantic knowledge representation standards to capture the case law data in a way that enables other related data to be semi-automatically linked with it.

From the above, we derive the need to develop software that: (1) is more user-friendly and accessible for the general legal scholar; (2) performs advanced visual analysis of case law integrated from multiple national databases; (3) is open-source, FAIR³³ and designed for both researchers and students alike; and (4) generates insights that are reproducible and shareable with other researchers and students. A tool that is open-source would allow continual improvement and addition of features by others in the legal research community.

Another important limitation of existing non-commercial software is that it does not attempt to integrate information from databases that fall outside the legal domain. Connecting information across different domains of knowledge can be relevant for answering legal research questions. For example, socio-economic and ecological data for EU member states may be used to answer interdisciplinary research questions that concern the law and its impact (see Figure 5). Figure 5 depicts the integration of information from a legal database (EUR-Lex) consisting of legislation

³³ Findable, Accessible, Interoperable and Reusable; MD Wilkinson et al, “The FAIR guiding principles for scientific data management and stewardship” (2016) *Scientific data* 3.

and court decision metadata; a prominent global database of statistics about socio-economic conditions in geographic locations around the world (World Bank³⁴); and a global database of statistics about greenhouse gas emissions (Emissions Database for Global Atmospheric Research (EDGAR³⁵)), in order to answer an interdisciplinary question concerning legal factors contributing to greenhouse gas emissions in densely populated EU countries.

IV. PROPOSING A RESEARCH ENGINE FOR LEGAL DATA ANALYTICS

In this section, we explore what a software platform designed to fulfil the needs identified in the previous section might look like. We discuss the major functional and design requirements of a software research engine for publicly available judicial data, and possibly additional data. We do so by providing examples and proposals that are understandable for both legal scholars and computer scientists.

1. Design requirements

Data infrastructure: the “fuel” of the proposed research engine is publicly accessible Linked Open Data (LOD)³⁶ about judicial data. Linked Data, as mentioned in Section 3, is information represented on the Web in a machine-readable and human-readable format that is recommended by the standards body of the World Wide Web (ie the W3C). LOD refers to Linked Data that is freely and publicly accessible on the Web. Currently there are 1,234 LOD datasets on the Web covering information from a wide variety of domains.³⁷

The EUCases project has been instrumental in converting case law information in the field of consumer law from multiple public databases across the EU into the LOD format. As part of the project, case law from databases in Germany, the UK, France, Bulgaria, Italy and Austria were converted and made publicly available.³⁸ To build upon this seminal work, we propose to enrich this information with case law from other legal domains and other EU member states such as Finland,³⁹ which have already begun conversion of their case law to the Linked Data format. This project involved several components. Firstly, relevant information about cases was gathered from documents published on different websites from different levels of Finnish courts. The formats of these documents varied from HTML to XML and PDF. The metadata of these documents were represented using multiple controlled vocabularies or thesauri. Therefore, these thesauri were first harmonised so that different terms describing the same content (across the levels of Finnish courts) were mapped to each other. Thereafter, these vocabularies were enriched with new terms (using the RDF standard) required to describe additional content of the documents. A text annotation

³⁴ <data.worldbank.org>.

³⁵ <edgar.jrc.ec.europa.eu>.

³⁶ <linkeddatabook.com/editions/1.0>.

³⁷ <lod-cloud.net>.

³⁸ <eucases.eu>.

³⁹ MFrosterus et al., “Facilitating Re-use of Legal Data in Applications-Finnish Law as a Linked Open Data Service” in *Proceedings of the International Conference on Legal Knowledge and Information Systems (JURIX 2014)* pp 115–124.

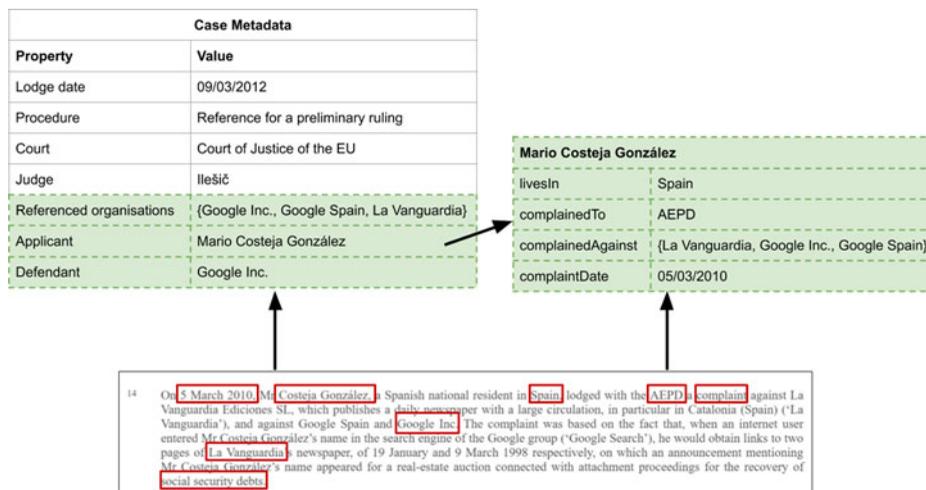


Figure 6: A conceptual illustration of how structured information can be obtained from unstructured court decision text, using NLP

tool was developed (called AATOS⁴⁰) to attach terms from the enriched vocabularies to content in the legal documents. Finally, the annotated information was translated into RDF format using automated computer scripts.

The Linked Data version of the EUCases project data primarily describes metadata about the cases such as the lodge and decision dates, unique case codes, judges, ruling types etc. We propose to add to this other relevant properties about the case that we obtain from its *content* ie the full text of the decision. This information may include items such as the organisations and parties involved, the specific topics discussed, and the main points of the decision. The EUCases project produced software which can automatically recognise such entities in case law text and annotate these with shared vocabulary about general legal terms (from the EUROVOC⁴¹ shared vocabularies website). This information should be added to the repository of the research engine. However, before it can be included for a particular case, it must first be extracted from the case text, validated by human experts, and integrated into existing Linked Data about this case. This data enrichment step should be performed on all relevant cases before the information is ingested into the research engines' data repository (see Figure 6 for an example illustration of the proposed enrichment). The idea conveyed in Figure 6 is that information in the case text itself can be extracted and structured as new metadata properties which can be added to existing metadata about the case. For example, the mentioned organisations in the case can be extracted using NLP and a property, say "referenced organisations", can be added to the metadata for the case, with the extracted values listed (for the case illustrated in Figure 6, this happens to include Google Inc, Google Spain, and La

⁴⁰ M Tamper et al, "Aatos – a configurable tool for automatic annotation" in *International Conference on Language, Data and Knowledge* (Springer 2017) pp 276–289.

⁴¹ <publications.europa.eu/en/web/eu-vocabularies>.

Vanguardia). Similarly, we can extract information about the applicant mentioned in the text (also using NLP). In Figure 6, we see that the place of residence of the applicant is extracted, and information about a complaint that the applicant lodged. The extraction and attachment of this structured information to the case allows deeper network analysis. Whereas previously one could “zoom in” to cases in the citation graph that involve a certain judge or that were lodged before a certain date (metadata that are already readily available in case law databases), with the additional extracted metadata one can be even more specific with filtering cases. For example, a user can analyse the citation graph of only those cases that involve complaints lodged with a specific organisation.

Publicly accessible case law metadata is usually available in a variety of data formats such as CSV and XML. XML in particular is a common format in case law databases such as EUR-Lex and Rechtspraak.nl. However, there are several naming conventions used to describe case metadata properties within the provided XML files. Akoma Ntoso⁴² and CEN Metalex⁴³ are two such standards for representing metadata of legal documents (including court decision texts). Both standards are translatable into Linked Data format using technologies such as RDF Mapping Language (RML).⁴⁴ Ideally, extraction, enrichment, conversion and cleaning of all the relevant case law data would be completed before the data is imported into the research engine. However, since the body of European case law is dynamic and continually expanding, having an automated or semi-automated tool to extract relevant information from case text and convert them to Linked Data format is a useful feature to include in the proposed research engine.

The Linked Data format is particularly suitable for the proposed software because it eases compliance with the FAIR⁴⁵ principles of data management. FAIR advocates that the persons responsible for generating and managing data should make clear all steps of their data management process so as to make it easier for other users downstream to reuse their data for other purposes (should this be permitted by the relevant stewards of the data). If data cannot be made publicly available, this should be made clear using relevant standards such as data licensing and disclosure terms. If there are special circumstances or procedures that should be followed to obtain data, these should be clearly stipulated and documented so as to make it easier for people to obtain access. Finally, data is prone to quality issues.⁴⁶

When metadata of cases on EUR-Lex are authored, there are sometimes inconsistencies encountered. For example, the “country” metadata field for a case can have values varying between “NL”, “The Netherlands”, “Netherlands” and “Holland”. To a human interpreter this is not usually a problem, however, for a computer, all these values are completely distinct. In order to ingest data of the

⁴² M Palmirani and F Vitali, “Akoma-Ntoso for legal documents” in *Legislative XML for the Semantic Web* (Springer 2011) pp 75–100.

⁴³ R Hoekstra, “The MetaLex document server” in *International Semantic Web Conference* (Springer 2011) pp 128–143.

⁴⁴ <rml.io>.

⁴⁵ Wilkinson et al, *supra*, note 33.

⁴⁶ AJ Zaveri et al, “Quality assessment for linked data: A survey” (2016) 7 *Semantic Web* 63.

highest possible quality into the software, we will perform data quality assessment to normalise such inconsistencies prior to importing.

User interface: the user interface of our proposed system is of paramount importance. It should make the analysis of judicial data truly accessible for the legal researcher, no matter their level of expertise. Thus the design of the system should revolve around this central theme of accessibility and usability. We propose to use state-of-the-art advances in user interface design to achieve this. The principle of minimalism (clutter-free interfaces) should, in particular, be adopted. This is because it has been shown that this heuristic for developing interfaces that are appealing and cognitively simpler for users to accomplish their tasks.⁴⁷ Nielsen's heuristics for interface design provide guidance as to how to develop a user interface design.⁴⁸

It is generally well-known that the use of visuals and media eases the learning process.⁴⁹ Since the main goal of the proposed research engine is to aid in the learning and analysis of case law, the interface should make full use of graphics and other visuals. Network analysis is a task that is inherently amenable to visual representation. However, while the graphical representation of cases in a network is helpful for quickly spotting patterns and relationships, in order to answer more subtle or complex questions, it may be required to analyse additional metadata about a specific case (or a cluster of cases). It may turn out to be impossible to represent all metadata for a case graphically (and in an aesthetically pleasing way) in the case citation graph. Even if one could do this, there is a risk of violating the usability guideline aimed at reducing the cognitive burden on the user.⁵⁰ Therefore, we propose to include a mechanism in the software to switch between the graphical view of the cases (and their relationships), and a tabular view of the information about the case. The tabular view should depict a table containing all metadata about *selected* cases (the user can select multiple nodes in the graph) in the network. Research has shown that offering such multiple views of a data source is cognitively beneficial for users.⁵¹ In fact, it has also been shown that switching between tabular and graphical modes of representation, for network analysis in particular, is beneficial especially when graphs become very large and one needs to separate and analyse distinct partitions of information in the graph.⁵²

The start page or entry point for the user can be a text box for specifying natural language questions (eg legal research questions about case law – see Figure 7).

We strongly recommend that the design of the interface be developed in consultation with the target users of the system (legal researchers and students).

⁴⁷ J Nielsen, "10 usability heuristics for user interface design" (1995) 1 Nielsen Norman Group.

⁴⁸ *ibid.*

⁴⁹ G Salomon, "Media and symbol systems as related to cognition and learning" (1979) 71 *Journal of Educational Psychology* 131.

⁵⁰ Nielsen, *supra*, note 47.

⁵¹ MQ Wang Baldonado et al, "Guidelines for using multiple views in information visualization" in *Proceedings of the working conference on Advanced visual interfaces* (ACM 2000) pp 110–119.

⁵² M Freire et al, "ManyNets: an interface for multiple network analysis and visualization" in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (ACM 2010) pp 213–222.

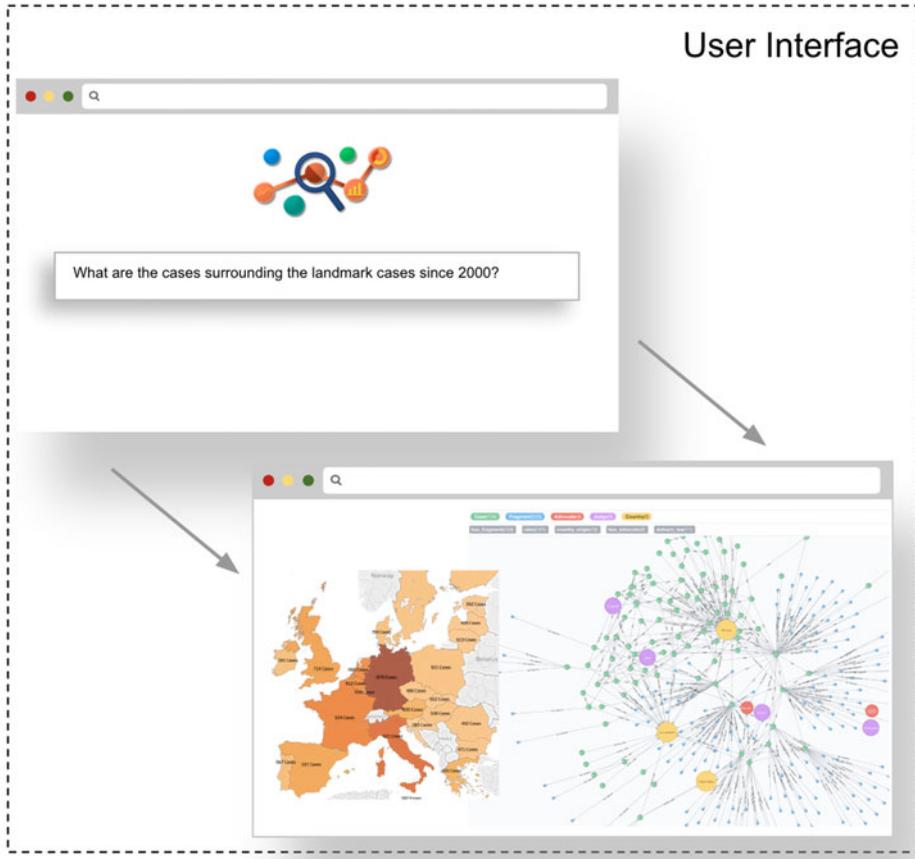


Figure 7: A possible look and feel of the proposed research engine's user interface

The recent advancement of technologies providing natural language query interfaces to structured data,⁵³ and state-of-the-art graph database technologies (eg D3.js⁵⁴) are key areas which have the potential to realise the vision of such a user interface. The envisioned mechanism for answering natural language questions will rely on converting the question to an intermediate computer-based format. In our situation, since the data about each case will be represented in RDF format, we plan to first use NLP to identify the named entities in the question (court names, judge names, dates etc). Thereafter, we can immediately query our data (using the RDF query language – SPARQL, which is an SQL-like language for querying information represented in Linked Data format⁵⁵) to identify what category each

⁵³ S Ferre, “Sparklis: an expressive query builder for sparql endpoints with guidance in natural language” (2017) 8 *Semantic Web* 405; A Styperek et al, “Semantic search engine with an intuitive user interface” in *Proceedings of the 23rd International Conference on World Wide Web* (ACM 2014) pp 383–384; V Tablan et al “A natural language query interface to structured information” in S Bechhofer et al (eds), *The Semantic Web: Research and Applications* (Springer 2008) pp 361–375; A Ngonga Ngomo et al, “Sorry, I don’t speak sparql: translating sparql queries into natural language” in *Proceedings of the 22nd International Conference on World Wide Web* (ACM 2013) pp 977–988.

⁵⁴ NQ Zhu, *Data visualization with D3.js cookbook* (Packt Publishing Ltd 2013).

⁵⁵ <www.w3.org/TR/sparql11-query>.

entity belongs to. Eg a particular court or judge will be mapped to the types “Court” and “Judge” respectively. We can then analyse the RDF data to identify specified relations between the entities mentioned in the question, which will help to answer the question by formulating SPARQL queries. This method is similar to the one applied by the LODQA tool used in biomedicine.⁵⁶

Application layer: The application layer should provide infrastructure for converting natural language questions posed by users (eg what are landmark cases for topic X?), to the standard W3C recommended SPARQL language. Existing technologies (such as RML⁵⁷) can be integrated in this layer of the software to convert additional case law data from traditional data formats such as CSV, XML and JSON to Linked Data format. The converted data can be stored in a database (eg a graph database⁵⁸). The infrastructure for managing this database, including the creation, editing, updating and deleting of case law data, can either be managed via third party software such as OntoText’s GraphDB⁵⁹ or via an extra software module as part of the research engine’s application layer. Container technology (eg Docker⁶⁰) may be used to package the system so that it can be deployed on the computer infrastructure of the user’s own institution. This feature would enable users from other universities and research institutions to host a copy of the software on their own computer infrastructure so as not to rely on a single copy to manage the demands of all users of the research engine. Docker solves the “works on my computer, but not on yours” problem by packaging all the necessary software dependencies and operating system resources required by an application in one independent software “container” that runs in exactly the same manner on any operating system and computer platform (eg Windows, Mac, Linux etc).

Furthermore, a RESTful Application Programming Interface (API) should be provided to make the data available to other software developers. An API is an online computer program that provides other computer programs with access to data. Very often, documentation for APIs (the “menu” for the types of data that can be accessed via the API) are poorly written and difficult to interpret by software developers who need to access and process this data.⁶¹ To improve the FAIRness of the API, one should provide documentation for the API that meets community standards of quality. For this purpose, the smartAPI⁶² specification may be used for documenting APIs in a FAIR manner. Figure 8 gives an overview of the proposed architecture.

Semantic layer: finally, in order to facilitate more advanced computational analysis of the court decisions in the citation graph, one should include algorithms that extend traditional network analysis techniques applied to case law. In particular, algorithms that automate the identification of semantic connections between the *content*

⁵⁶ KB Cohen, and JD Kim, “Evaluation of SPARQL query generation from natural language questions” in *Proceedings of the Joint Workshop on NLP&LOD and SWAIE: Semantic Web, Linked Open Data and Information Extraction* (2013) pp 3–7.

⁵⁷ <rml.io/spec.html>.

⁵⁸ R Angles and C Gutierrez, “Survey of graph database models” (2008) 40 ACM Computing Surveys 1.

⁵⁹ <graphdb.ontotext.com>.

⁶⁰ <www.docker.com>.

⁶¹ G Uddin and MP Robillard, “How API documentation fails” (2015) 32 IEEE Software 68.

⁶² See <smart-api.info>; L Richardson and S Ruby, *RESTful web services* (Reilly Media 2008).

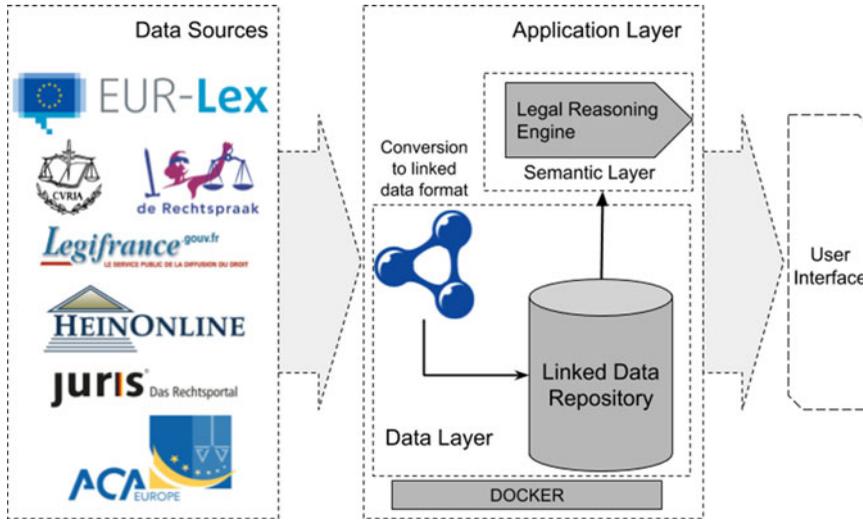


Figure 8: An overview of the technical architecture of the proposed research engine. The list of data sources is indicative but not exhaustive

(statements made) in different decisions should be developed. Artificial intelligence (AI) can be applied to achieve this. In particular, there have been substantial advances in AI measures for semantic similarity of texts⁶³ and this has potential to be used in the tool to recommend similar cases to legal scholars based on textual and argument similarity across cases. In the interests of aiding the learning process, it is recommended that algorithms be developed that are able to provide a traceable record of the steps applied, so as to enable transparent explanation to the users about how certain relations were identified. However, it remains challenging to provide fully explainable solutions when certain AI techniques are used – for example *deep learning*.⁶⁴

To recap, architecturally speaking, the proposed research engine should be composed of a user interface (handling user input of queries, as well as display and visualisation of data views), a “data layer” which houses the case law information analysed by the system, a so-called “application layer” (housing the main functional components of the software for retrieving, manipulating and processing case law data) and a “semantic layer” (the algorithms which perform inference and advanced analysis on the data).

2. Functional requirements

Not all users of the research engine would need to analyse the same cases. Some may focus on a certain national database (eg *Rechtpraak.nl* in the Netherlands), some may need to analyse connections between European-level court decisions and national databases (eg between cases in EUR-Lex and those in Austria’s RIS), and others may only need to analyse cases based on a specific topic relevant for a particular legal field (eg competition law). Therefore, one of the major functional requirements is to

⁶³ O Levy et al, “Improving distributional similarity with lessons learned from word embeddings” (2015) 3 *Transactions of the Association for Computational Linguistics* 211.

⁶⁴ Y LeCun et al, “Deep learning” (2015) 521 *nature* 7553 at p 436.

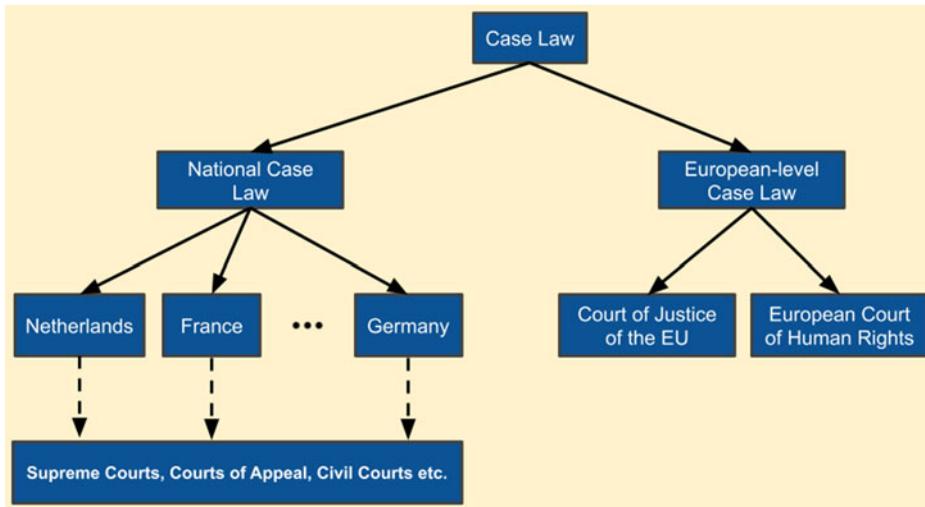


Figure 9: High-level categories of case law that should be included in the research engine

be able to filter the relevant subset of cases for the analysis. Users should thus be permitted to either perform “drill-down” search by clicking on nodes in a graph representing specific subdomains of law (eg competition law), or via typing of natural language questions into a search box (see for example Figure 7).

The former strategy may require more clicks to get to the relevant subset of cases, whereas the latter is more difficult to implement from a technological perspective because understanding natural language is still challenging for computers. We advocate that both strategies could be useful for this purpose, and that they both should be explored and researched further. Using these methods, the software can locate the required level of specificity for the relevant subset of case law (see Figure 9 for the high-level categories of case law). The user can then be presented with a data analytics dashboard for exploring and visualising the data with graphs. The dashboard is a user interface that will be composed of two “sections” in the interface, one for conducting qualitative analytics, and the other for quantitative analytics.

Qualitative dashboard: the function of the qualitative dashboard would be to conduct advanced network analysis on cases. This part of the research engine should be tailored to answer questions about the content overlap between cases, detection of citation communities, to identify landmark decisions etc. The components of this section should be:

- *A network analysis graph:* of the dataset that has been isolated by the user’s query. This would be a subset of cases in the graph database relevant to the user’s query.
- *Faceted search controls:* faceted search is a technique in user interfaces allowing users to narrow down their search results by applying a variety of filters or criteria to their search. This would allow filtration of the nodes (cases) in the graph according its properties (eg topic, presiding judge etc). The controls would be analogous in functionality to the panel depicted in the left hand region of the interface in Figure 3. However, the filters should be grouped according to their

similarity to make them easier to locate. The controls should also be designed in a more visually appealing way, and positioned with more space between them. Each filter should provide descriptions in the form of tooltips to enable the user to understand what they represent.

- *Data download controls*: these controls may be implemented as links to allow exporting of the filtered dataset in standardised formats such as CSV, TSV, JSON and Linked Data. This would enable users with more technical proficiency to import information into other software, should they need to, for other kinds of processing not offered by the research engine (this feature promotes the software's interoperability).
- *Toggle views feature*: ie a control to toggle the view of the data between a graph-based view and tabular-based view. The graph-based view is useful in order to gain a high-level view of the connections between cases. In some situations, in order to be more user-friendly and visually appealing, information has to be hidden in the graph-based view. However, if a researcher would like more detailed information about the underlying data that the graph is based on, they can switch to a tabular view. The tabular view should also provide provenance information about the data (where it was extracted from, when, and what processing methods were used on them, if any). The provenance is critical for determining the veracity of the data, which in turn affects the veracity of the conclusions we can draw from analysing the case law graphs. The provision of provenance for data also supports the reusability principle of FAIR because it allows the user to make an informed decision about the quality of the data and its suitability for a certain analysis.
- *Controls to automatically share analyses with other researchers*: one of the major motivations behind the FAIR principles is to promote reproducibility of data analyses. In scientific fields such as biomedicine, reproducibility of other researchers' analyses is a major problem.⁶⁵ This, of course, makes it difficult to independently verify the accuracy of claims and assertions made. In the interests of avoiding this problem in empirical legal research, there has to be emphasis on enabling reproducibility of analyses. Transparency of which case law databases were used in the analysis is one factor to address towards this. Another is the documentation and sharing of the specific data processing and analysis steps used to reach the results and conclusions. We propose that the research engine should record and store the sequence of user interactions submitted to the engine that produced the current view for the user. We also advocate that there be a feature in the software that allows one user to share this exact sequence of interactions automatically with other users who may be interested (either via email or across accounts on the engine itself, for example).

Quantitative dashboard: the role of the quantitative dashboard is to provide tools for *descriptive statistics* (at the minimum) about the selected case law. This section can aid users in answering more quantitative questions about cases such as "What is the most

⁶⁵ CG Begley and JP Ioannidis, "Reproducibility in science: improving the standard for basic and preclinical research" (2015) 116 *Circulation Research* 116.

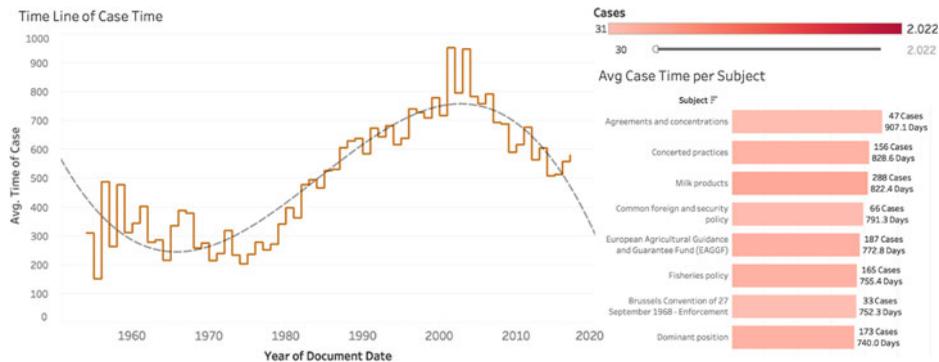


Figure 10: An example plot that might be displayed in the quantitative dashboard. This particular graph displays the average duration of sample cases decided by the Court of Justice of the European Union (extracted from the EUR-Lex database)

cited case and paragraph in the Court of Justice of the EU?”, “How long does it take on average for Judge Smith to decide his cases?”, “How many cases have emerged in the last five years on gender equality in the European Court of Human Rights?”. There should be controls in this dashboard for the user to generate graphs and plots to answer questions like the ones mentioned above (see Figure 10 for an example plot). The required controls we suggest are as follows:

- *Plot area:* this area will contain a set of plots or graphs that describe statistics and properties about the specific case law dataset selected.
- *Metadata list:* there will be a panel to the left of the plot area which contains a list of metadata about the case. These could be “dragged and dropped” onto a plot in order to generate a graph describing the relationship between the properties in the dataset. For example, to see a plot about the duration of cases per judge, the user would drag the “judge” and “case duration” properties onto a plot.
- *Plot type selectors:* to give the user control of the visual type of the graph to generate, there should be a control to the right of the plot area which allows the user to select this type: eg line, bar or histogram, pie-chart.
- *Range filters:* to enable the user to plot only a certain range of values for certain metadata fields (eg they only want to see the case duration for certain judges, or for cases lodged in a time range like 2014–2018), there should be controls displayed on each plot that allow the user to adjust the desired range of values. The plot should dynamically change to reflect the new selection of values in real-time as the controls are adjusted.

V. CONCLUSION AND FUTURE WORK

We have suggested the development of a software research engine that facilitates user-friendly quantitative and qualitative data analytics on legal linked open data, for legal scholars with limited technical expertise. We have also discussed the design and

functional requirements that such software should offer to be useful to the general legal scholar.

So far, by and large, software that enables advanced legal data analysis has been dominated by the commercial sector. Their pricing, focus on legal practice, and the data science proficiency required to use them, leaves a gap for the development of alternative software that is FAIR, publicly available, open-source, and easy to use for a general legal scholar. Software research engines like the one proposed in this paper have the potential to aid legal scholars in conducting impactful empirical legal research. This is very necessary, since, contrary to popular belief and despite the rapid advancement in data science methodologies and computer hardware, empirical legal research in Europe has not been rising in prevalence in legal journals.⁶⁶

The envisioned system could also be used as a training tool to educate students about the basic principles behind this kind of research, without waiting for the current legal educational programmes to be augmented with empirical and data science methodologies (although empirical training is obviously preferred). Additionally, the existence of a software platform could accelerate empirical training of legal scholars, and empirical legal research in general. Regardless, the increased availability of digitised legal texts and the advancement of data science methods offers potential, to the point that they are able to assimilate vastly larger quantities of legal texts than any human scholar could at any given time.

We suggest the consolidation of the detailed design of the research engine presented in this paper, together with the help of its target users (legal researchers and students). Detailed design decisions need to be made for both the user interface, and the use cases (functionality) required by the potential users. We also recommend the collection, and performance of data quality assessment⁶⁷ on, all official and publicly available case law data on the Web. This is with a view to providing datasets in the research engine that are free from major data quality inconsistencies which could reduce the validity of the analyses generated with the software. Finally, we recommend that a prototype of the software be created, and its usability be evaluated using state of the art software usability testing methods.⁶⁸

⁶⁶ G van Dijck et al, "Empirical Legal Research in Europe: Prevalence, Obstacles, and Interventions" (2018) 11 *Erasmus Law Review* 105.

⁶⁷ Zaveri et al, *supra*, note 46.

⁶⁸ F Paz and JA Pow-Sang, "A Systematic Mapping Review of Usability Evaluation Methods for Software Development Process" (2016) 10 *International Journal of Software Engineering and Its Applications* 165.