LEXIA: A DATA SCIENCE ENVIRONMENT FOR SEMANTIC ANALYSIS OF GERMAN LEGAL TEXTS

Bernhard Waltl / Florian Matthes / Tobias Waltl / Thomas Grass

Research Associate, b.waltl@tum.de, Technical University of Munich, Department of Informatics,
Software Engineering for Business Information Systems, Boltzmannstraße 3, 85748 Garching bei München, DE

Professor, matthes@tum.de; https://wwwmatthes.in.tum.de/

Students, Information Systems, {waltlt, grasst}@in.tum.de

Keywords: ***Legal Data Science, Semantic Analysis, Text Analysis, Apache UIMA, Apache Ruta***

Abstract: *The analysis of legal data using information technology, more specifically text and data mining algorithms, has become very attractive in the field of legal informatics. Additionally, legal science and practice consist of data-, knowledge-, and time-intensive tasks, which have always been in the focus of legal informatics. This paper contributes a data science environment, which is in particular suited for legal texts, e.g. documents from legislation and jurisdiction but also contracts and patents. The environment consists of a reference architecture and a specific data model. Furthermore, it integrates an easily adaptable and extendable text mining engine allowing reuse of components. The base line architecture for the text mining engine is the Apache UIMA. The environment enables to collaboratively specify linguistic and semantic structures. Thereby, it uses an existing rule-based script language, namely Apache Ruta.. This paper shows how the system can be used to unveil legal definitions in the German Civil Code (BGB) by not only finding them but also by determining which legal term is defined and how. This functionality enables the structuring of unstructured information, i.e., text, which enables data scientists and legal experts to semantically investigate and explore legal texts.*

1. Introduction

Recent developments in legal science and practice have shown that legal data analysis is a promising field. Legal tasks are increasingly becoming data-, time-, and knowledge-intensive. On the other hand, computer science has made huge progress in the domain of data mining, in particular in text mining. Algorithms processing unstructured information, i.e. text, can produce highly accurate results, with respect to precision but also to recall. Although the algorithms have been developed and are continuously improving, only less effort has been spent on tailoring those technologies to the legal domain. However, this tailoring is a crucial step in order to reproduce those high quality results outside the domains for which those text mining technologies have been developed and trained. This paper proposes a data model, which was developed to represent the particularities of legal literature (see Section 1). Thereby, it allows the processing, generation, and persisting of structured but also of unstructured data. The implementation of an adapted UIMA (Unstructured Information Management Architecture), which has originally been developed by IBM and was later on the base line for the IBM Watson software suite, allows the development and usage of most recent natural language processing (NLP) technologies. Furthermore, it enables the integration of components to perform qualitative and quantitative analysis on text corpora (see Section 5). Based on that, huge legal text corpora can be analyzed, summarized, explored and visualized in a data scientifically way. This allows the creation of new views, perspectives and representations on textually represented data. This can exemplarily be used to determine semantic structures in laws or to semantically analyze textual norms within claims examinations or due diligence to unveil incon-

sistencies or vagueness, increasing the exposure to risk. It has the potential to be a paradigm shift in legal science and practice.

2. Research method and objectives: The legal domain – a challenge for data science

To achieve highest accuracy with NLP it is necessary, that the environment and data model are adapted to the domain of investigation, which is in this case the domain of legal data modelling and processing with a special focus on Germany. However, much effort has already been spent on NLP and processing of unstructured data (see Section 3) but hardly anyone investigated the environment which can be the base line for the analysis and allows for reuse and interoperability of components. By providing a reference architecture and a generic and flexible data model this paper shows how a tailored base line environment can look like.

Throughout this paper an adapted design science approach, originally proposed by Hevner (2004), has been used. Firstly, the problem relevancy has been analyzed and is briefly sketched (see Section 1 above and Section 3). Secondly, the focus is set on the design of a new artifact, i.e., a data science environment (see Section 4). Afterwards, the applicability is evaluated and the research contribution is summarized (see Sections 5 and 6).

3. Related work

In 2010, de Maat and Winkels (2010) have published an article on the classification of norms. Thereby, the classification was based on work by Hart (1961). By creating a layer model, they differentiated between four level of norms. This model served as the baseline for automated classification, which they performed with a Java pattern matcher (Maat and Winkels, 2010, pp. 182). Using regular expressions, they were able to classify 91 % of all phrases correctly. However, they admit that the expressions have to be improved, especially when it comes up to the consideration of linguistic properties, like auxiliary sentences. It is unlikely that this can solely be achieved with regular expressions. Furthermore, regular expressions are hard to maintain and can hardly be composed and reused, which is a major drawback of this approach.

Bommarito and Katz (2014) focused on the analysis of the US Code's complexity based on multiple dimensions. Aiming at a description of the text complexity, they differentiate between structural, such as hierarchies and network-like dependencies, and semantic properties of words and sentences. Finally, they aggregated the structural, linguistic and semantic properties to retrieve an overall complexity measure. As most recent publications at relevant scientific conferences show, network analysis, similarity measurements, and dependency investigations in norm corpora are becoming increasingly relevant.

A further relevant field for applying natural language processing techniques addresses the reconstruction and automated extraction of arguments from texts, especially legal cases (Walter, 2009; Wyner, 2010). The extraction of arguments in cases is mainly done by determining the linguistic structure and argumentative dependencies (Gordon, 2007). With increasing regularity, the accuracy of the results increases. A paper published in 2015 by Houy et al. describes the attempt of mining argumentation patterns from decisions within the German Federal Constitutional Court (Houy, 2015). They identified several structures indicating an argument, which they extracted from supreme court decisions using standard NLP modules namely tokenization, parts-of-speech tagging, stemming, and named-entity recognition.

Recently, Grabmair et al. (2015) published preliminary results about the adaptation of the UIMA[1]. They developed a «law-specific semantic based toolbox» to automatically annotate semantic patterns at a sub-sentence level. The rules expressing those phrases are codified in Apache Ruta[2], which is also used in data science en-

[1] Apache UIMA, https://uima.apache.org/, all web pages last access on 12 December 2015.
[2] Apache Ruta, https://uima.apache.org/d/ruta-current/tools.ruta.book.html.

vironment proposed in this article. Apache Ruta is not only an established but also a maintainable and reusable pattern definition expression language.

As stated above, text mining and algorithmic processing of textual data has been in the focus of researchers and generic but monolithic workbenches, e.g., GATE (with its Jape grammar), and software packages, e.g., NLTK, have already been proposed. However, without a particular tailoring to the legal domain those do not enable data scientists to easily produce most accurate results and additionally allow the interaction with visualization and reporting engines for discovering and exploring algorithmically processed legal data.

4. Reference architecture and data model

4.1. Reference architecture for the data science environment

Figure 1 shows a comprehensive reference architecture consisting of an importer, an exporter, a data storage and access layer, a text mining engine, and a user interface. Based on the reference architecture we developed a collaborative web application with a Java back-end. Additionally, the search engine ElasticSearch for efficient access to the textual data has been integrated.

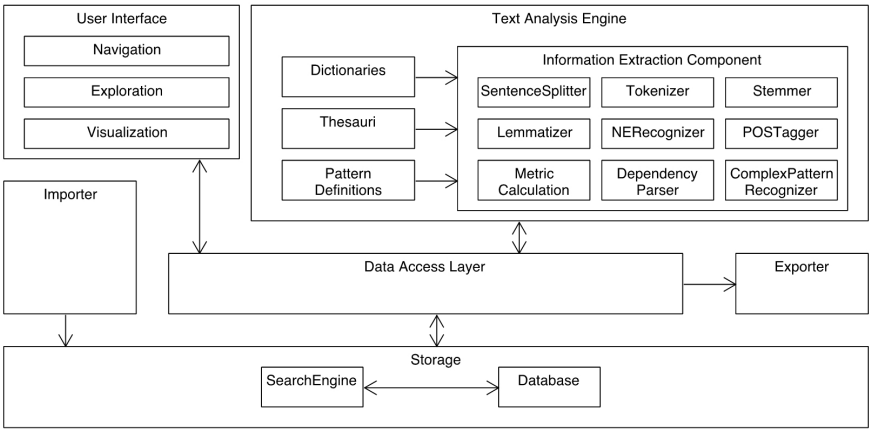

Figure 1: Legal data science reference architecture for collaborative environments

4.2. Data Model, Data Storage and Access

Figure 2 shows the data model that is used to represent the laws and the judgments. For sake of simplicity concrete attributes, properties, and methods of the classes are omitted. The concrete attributes may vary throughout the implementation and the particular use cases. However, within the implementation relevant values are stored in Key-Value-Maps, which do not constrain the usage of attributes beforehand. The *LegalDocument* class is the base class for the different types of documents, e.g. laws or judgments. Each *LegalDocument* has a class *Metadata* attached, which holds additional information and attributes for a document. To add a new type of document, such as *Contract*, the *Contract* has to be implemented as a derived class (e.g., *Judgment*) from *LegalDocument*. The new class not only reuses the remaining nested data structure but also the algorithms and information extraction components.

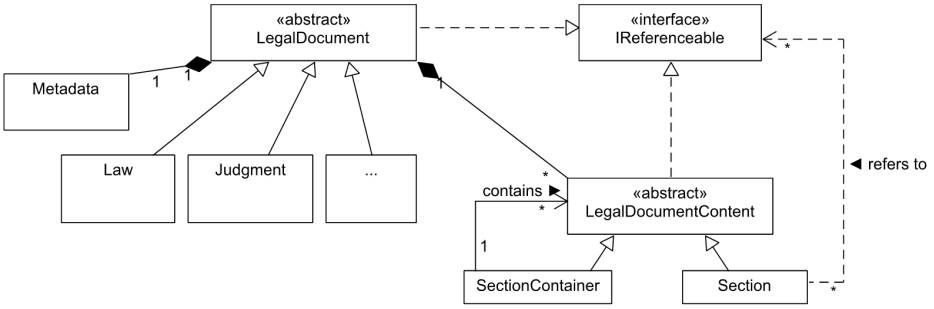

Figure 2: Conceptual model for the internal representation of legal data

The content of the *LegalDocument* is implemented as a composite pattern (component: *LegalDocumentContent*; composite: *SectionContainer*; leaf: *Section*) (Gamma, 1994), this follows the nested structure laws in Germany have. Therefore, the data model is able to represent the nested structure of laws or judgments but is also open for flat hierarchies in documents. E.g. the German Civil Code consists of two books, which consist of several divisions, which – in turn – consist of several titles and so forth (German Civil Code, 2014). At the end of this different containers, the *Sections*, which hold the content of norms, are stored. This tree like structure has to be conserved by the data model since it is part of official legislative and jurisdictional documents and the containers group logically related norms together, which can then be referenced. The referencing structure is implemented via interfaces, such that each *Section* can have multiple references to *LegalDocument* or to *LegalDocumentContent*.

4.3. Text Mining Engine

Based on the data model the text mining engine applies state-of-the-art analysis methods to legal texts. It consists of a variety of reusable components and can easily be extended with new components. As baseline architecture Apache UIMA was used. UIMA allows the composing of components in a generic and easy adaptable environment. Furthermore, it is suitable for an integration in a collaborative web platform. Consequently, the *Information Extractor Component* (Figure 1) extracts and annotates semantic information a legal text. Additionally, dictionaries, thesauri, and pattern definitions are required in order to detect this semantic information accordingly. We are able to use two kinds of pattern definitions: regular expressions (regex) and Ruta expressions (rule-based text annotation). The Ruta expressions allow the specification of complex linguistic structures (see Listing 1).

4.4. Importer and Exporter

The importing structure is required to transform the input data, which can be of any data type (PDF, Word, XML, etc.), into the data model of our system. Therefore, the import architecture was designed such that it is possible to develop highly specified import components and at the same time easily support new file and document types.

The exporter component provides interfaces for other applications (e.g., REST APIs). The current implementation only provides methods to create data dumps (CSV) of the semantic entities and their occurrence, allowing the reuse in existing reporting and dashboard engines (e.g., Excel, etc.). Based on upcoming use cases the exporter component can be enhanced and adapted to support more functionality.

5. Unveiling semantic structures in laws

The determination of semantics arising within normative texts, e.g., laws, is known to be non-trivial. The interpretation of legal texts is complex and even in the field of legal theory controversially discussed. However, considering various hermeneutical and logical approaches, such as grammatical, systematical, historical or teleological interpretation (Canaris, 1994), the grammatical interpretation is closest to the concrete wording, e.g. the text itself. Therefore, and because the text is a direct declaration of the legislator's intent, it is rational to analyze the wording of the normative text. Necessarily, legal data analysis starts with the examination of semantics that arises from the linguistic information represented in the text. Furthermore, legal texts are well-known for their highly standardized usage of vocabulary and linguistic patterns. Based on this, we show how it is possible to reconstruct the semantics in legal texts by determining the linguistic patterns and annotating the legal text accordingly.

5.1. Determining legal definitions in legal texts using Apache UIMA and Apache Ruta

```
1  // Basic linguistic vocabulary
2  DECLARE ISDG;
3  "im Sinne dieses Gesetzes" -> LDSache.ISDG;
4  "im Sinne des Gesetzes" -> LDSache.ISDG;
5
6  DECLARE IST;
7  "ist|sind" -> LD.IST;
8
9  DECLARE NEG;
10 "keine|kein|nicht" -> LD.NEG;
11
12 DECLARE LDIdentifier; // Declare the indicator for legal definitions
13 DECLARE LegalEntity; // Declare the legally defined entity
14 DECLARE LegalDefinition; // Declare the legal definition
15
16 // Definition of linguistic patterns and rules
17 // {{ADJ}} {{NOUN}} im Sinne dieses|des Gesetzes ist {{Phrase}}
18 ((pos.N? pos.N) {-> LD.LegalEntity} LD.ISDG) {-> LD.LDIdentifier};
19 ((pos.ADJ+ pos.N) {-> LD.LegalEntity} LD.ISDG) {-> LD.LDIdentifier};
20
21 // {{NOUN}} ist kein {{NOUN}}
22 (pos.N {-> LD.LegalEntity} LD.IST LD.NEG pos.N) {-> LD.LDIdentifier};
23 (pos.N{-PARTOF(LD.LegalEntity) -> LD.LegalEntity} LD.ISDG){->LD.LDIdentifier};
24
25 // Annotate the sentence being a legal definition as LegalDefinition
26 Sentence{CONTAINS(LD.LDIdentifier) -> LD.LegalDefinition};
27
28 // Remove temporary annotations
29 LD.IST {-> UNMARK(LD.IST)};
30 LD.NEG {-> UNMARK(LD.NEG)};
31 LD.ISDG {-> UNMARK(LD.ISDG)};
32 LD.LDIdentifier{-> UNMARK(LD.LDIdentifier)};
```

Listing 1: Linguistic pattern descriptions (LD.ruta) for the semantic entity Legal Definition using Apache Ruta

Listing 1 defines the linguistic pattern specifying legal definitions in the German Civil Code. Determining legal definitions is highly relevant in the legal domain and has been in the focus of researchers for several times (see Section 3). The code example starts by defining baseline linguistic patterns as words and key phrases (Line 1–10). Afterwards, it specifies the rules that, if found in the legal text, refer to a legal definition. For instance, the legal definition of the term «Sache» in the German Civil Code (BGB, §90ff):

§ 90: Eine Sache im Sinne des Gesetzes sind nur körperliche Gegenstände.

§ 90a: Tiere sind keine Sachen. [...]

§ 91: Vertretbare Sachen im Sinne des Gesetzes sind bewegliche Sachen, die im Verkehr nach Zahl, Maß oder Gewicht bestimmt zu werden pflegen.

§ 92: Verbrauchbare Sachen im Sinne des Gesetzes sind bewegliche Sachen, deren bestimmungsmäßiger Gebrauch in dem Verbrauch oder in der Veräußerung besteht.

The listing shows how the variations of the legal definitions can be handled, by taking into account existing annotations such as parts of speech and building complex patterns by reusing those existing annotations. The rule specified in line 19 matches all nouns with corresponding adjectives that are followed by a particular phrase, namely «im Sinne des Gesetzes» or «im Sinne dieses Gesetzes». Thereby, the noun and its adjectives will be annotated as *LegalEntity* and the complete pattern as *LDIdentifier*. Line 26 finally marks all sentences containing *LDIdentifiers* as *LegalDefinitions*. Finally, temporary annotations, which are no longer required, are unmarked.

5.2. UIMA pipeline for semantic annotation of legal texts

The pipes & filters architecture of UIMA allows the creation of processing pipelines (see Figure 3). Various tailored software components (Splitter, Segmenter, Tokenizer, Tagger, Ruta) are concatenated in order to fulfill a complex processing task. We have developed the components especially for the German domain, taking into account the textual and editorial style, such as abbreviations, enumerations and listings, bracketing, etc. They can be (re-)used individually and almost arbitrarily be combined for other texts of the German legislation.

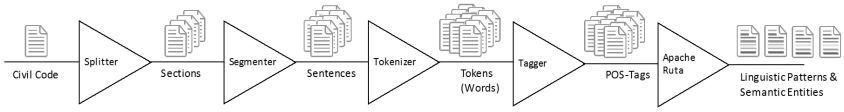

Figure 3: Processing pipeline for determining linguistic patterns with Apache UIMA and Ruta.

The last processing step of the pipeline shown in Figure 3 takes the script as introduced above into account and determines the semantic entities «Legal Definitions». Finally, those semantic entities are persisted and the user is not only able to export them into a separate CSV file, but also to access it via a search interface, view it in the browser instantly and share the view with other users collaboratively.

5.3. Accessing results and annotations through the user interface

The user interface allows the data scientist to access the textual representation of the law. After the processing of an analysis pipeline, the user does not only see the actual law text and its structure, i.e., books, chapters, subchapters, sections, etc., but also the available semantic entities that have been automatically determined. The exploration screen is divided into three different sections, namely the control section (left), the text section (middle) and the information section (right). Figure 4 shows the user interface. Thereby, the user interface is interactive and enriches the textual representation with the information selected by the user. Figure 4 shows the highlighted semantic entities «LegalDefinition» (dark green) and «LegalEntity» (turquoise).

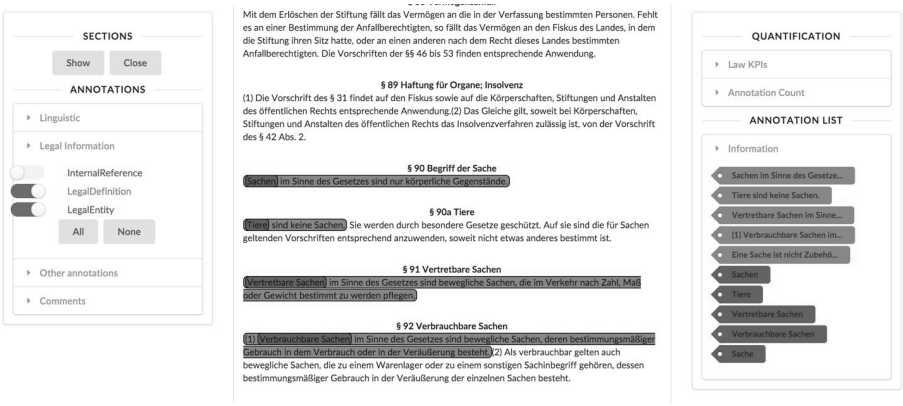

Figure 4: Interactive user interface and visualization of semantic entities

In addition to the highlighted passages in the text, the available annotations are listed in information section as it can be seen in Figure 4.

This functionality of the textual representation combined with the determination of semantic entities through linguistic patterns enables data scientists to interactively explore German legal texts. Within their exploration they are not constrained to full-text search or the integration of thesauri or ontologies, they are also capable of specifying and defining complex linguistic patterns, which are going to be automatically determined using reusable software components.

6. Conclusion, outlook, and future applications

This work is an additional contribution to the emerging field of legal data science, which is due to the advances in artificial intelligence and information retrieval becoming more and more popular. In addition, the computational power and availability of text mining algorithms, such as parts-of-speech taggers and linguistic parsers, make it attractive to analyze texts from the legal domain.

Although several attempts have already been made to adapt text mining algorithms for the legal domain, rather less effort has been spent on the design and tailoring of a data science environment for the German legislation. This is contra intuitive in several ways: to achieve highest accuracy in terms of precision and recall it is necessary to fully tailor the environment to the prevailing domain, i.e., legal, and to avoid the time-consuming repeated implementation of software components, which could be reused easily. This work now proposes a data science environment, which has been developed for the German legal domain. Thereby, its data model allows the representation of legal texts and their nested and referential structure. The core of this data science environment is the text mining engine, which was developed based on a state-of-the art pipes & filters software architecture, namely Apache UIMA (unstructured information management architecture). This architecture fosters the reuse of software components and allows an adaption to the particular domain of legal texts. We developed several of those software components for the analysis of the text. This is the baseline for highly accurate, algorithmic textual data analysis. In order to unveil the semantics of legal texts, represented through linguistic patterns, it is necessary to determine complex linguistic patterns, which can be done with our environment. Therefore, we integrated a rule-based expression language and exemplarily showed how legal definitions can be determined using this expression language. The data scientists access the results of the data analysis process through the web application itself. Thereby, the system highlights the patterns with different colors on the legal text itself, similarly to other tools. In addition, and to foster the analysis in other

tools and workbenches like R, Matlab, Excel, etc. the available annotations can be exported as CSV files.

We will continue the development of the prototype into three different directions, which will make it a more comprehensive tool for linguistic analysis of legal documents: at first, we will improve the visual representation of the annotations and the text. It is planned to make a more dashboard-like view on the document corpus, which allows the exploration. Secondly, we improve the integration of the semantic entities into the full-text search interface. Finally, we enhance the collaborative aspect of the web-based environment. This tool should be a workbench for data scientists and legal experts to share knowledge about linguistic patterns. The complex tasks of linguistically and semantically exploiting a legal text corpus can be done collaboratively to avoid resource-intensive reinvention and reimplementation of the same data models, algorithms and linguistics patterns.

7. Acknowledgement

This research was sponsored in part by the German Federal Ministry of Education and Research (BMBF) (project «Software Campus (TU München)», grant no. 01IS12057).

8. Bibliography

Larenz, Karl; Canaris, Claus-Wilhelm (1995): Methodenlehre der Rechtswissenschaft. Berlin, Springer.

Katz, Daniel; Martin; Bommarito, M. J., II (2014): Measuring the complexity of the law: the United States Code. In: *Artif Intell Law* 22 (4), pp. 337–374. DOI: 10.1007/s10506-014-9160-8.

Federal Ministry of Justice and Consumer Protection: German Civil Code. Hg. v. juris GmbH.

Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John (1994): Design patterns: elements of reusable object-oriented software: Pearson Education.

Gordon, T; Prakken, H.; Walton, D. (2007): The Carneades model of argument and burden of proof. In: *Artificial Intelligence* 171.

Grabmair, Matthias; Ashley, Kevin D.; Chen, Ran; Sureshkumar, Preethi; Wang, Chen; Nyberg, Eric; Walker, Vern R. (2015): Introducing LUIMA: An Experiment in Legal Conceptual Retrieval of Vaccine Injury Decisions Using a UIMA Type System and Tools. In: Proceedings of the 15th International Conference on Artificial Intelligence and Law, pp. 69–78.

Hart, H. L. A (1961): The concept of law. Oxford University Press (Clarendon law series).

Hevner, Alan R.; March, Salvatore T.; Park, Jinsoo; Ram, Sudha (2004): Design science in information systems research. In: *MIS quarterly* 28 (1), pp. 75–105.

Houy, Constantin; Niesen, Tim; Calvillo, Jesús; Fettke, Peter; Loos, Peter; Krämer, Annika et al. (2015): Konzeption und Implementierung eines Werkzeuges zur automatisierten Identifikation und Analyse von Argumentationsstrukturen anhand der Entscheidungen des Bundesverfassungsgerichts im Digital-Humanities-Projekt ARGUMENTUM. In: *Datenbank Spektrum*, pp. 1–9. DOI: 10.1007/s13222-014-0175-9.

Maat, Emile de; Winkels, Radboud (2010): Automated Classification of Norms in Sources of Law. In: Enrico Francesconi (Hg.): Semantic processing of legal texts. Where the language of law meets the law of language. Springer, pp. 170–191.

Walter, Stephan (2009): Definition extraction from court decisions using computational linguistic technology. In: *Formal Linguistics and Law* 212, p. 183.

Wyner, Adam; Mochales-Palau, Raquel; Moens, Marie-Francine; Milward, David (2010): Approaches to text mining arguments from legal cases. In: Enrico Francesconi (ed.): Semantic processing of legal texts. Where the language of law meets the law of language. Berlin, New York: Springer, pp. 60–79.