Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

BRISE-Plandok: a German legal corpus of building regulations

Gábor Recski (gabor.recski@tuwien.ac.at)
TU Wien
Eszter Iklódi
TU Wien
Björn Lellmann
TU Wien
Ádám Kovács
TU Wien
Allan Hanbury
TU Wien

Research Article

Keywords:

Posted Date: March 21st, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2717413/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

BRISE-Plandok: a German legal corpus of building regulations

Gábor Recski[1], Eszter Iklódi[1], Björn Lellmann[1,2], Ádám Kovács[1,3] and Allan Hanbury[1,4]

[1]TU Wien, Vienna, Austria.
[2]Federal Ministry of Finance, Vienna, Austria.
[3]Budapest University of Technology and Economics, Budapest, Hungary.
[4]Complexity Science Hub, Vienna, Austria.

Abstract

We present the BRISE-Plandok corpus, a collection of 250 text documents with over 7,000 sentences from the Zoning Map of the City of Vienna, annotated manually with formal representations of the rules they convey. The generic rule format used by the corpus enables automated compliance checking of building plans, a process developed as part of the BRISE* project. The format also allows for conversion to multiple logic formalisms, including dyadic deontic logic, enabling automated reasoning. Annotation guidelines were developed in collaboration with experts of the city's building inspection office, describing nearly 100 domain-specific attributes with examples. Each document was annotated independently by two trained annotators and subsequently reviewed by the authors. A rule-based system for the automatic extraction of rules from text was developed and used in the annotation process to provide suggestions. The reviewed dataset was also used to train a set of baseline machine learning models for the task of attribute extraction, the main step in the rule extraction process. Both the rule-based system and the ML baselines are evaluated on the annotated dataset and released as open-source software. We also describe and release the framework used for generating and parsing the interactive xlsx spreadsheets used by annotators.

*https://smartcity.wien.gv.at/en/brise/

1 Introduction

The majority of legal adjudication is in many countries an administrative process, typified by routine decisions on what is admissible based on legal text and legal precedent. The high volumes of such administrative adjudications can lead to backlogs, inconsistencies, high resource loads for government departments, and uncertainty for citizens (Branting, Yeh, Weiss, Merkhofer, & Brown, 2018). The city of Vienna, Austria, initiated the BRISE project as part of an e-government initiative to improve the efficiency of adjudication processes. The BRISE project is overall concerned with transforming the process of applying for building permission from a paper-based submission process to a digital submission process, in which digital plans of buildings are submitted instead of paper print-outs. Part of the project, considered in this paper, involves developing a decision support system for the building inspectors based on automated compliance checking of building plans.

Beyond the Vienna building code, there exist around 1,000 documents describing exceptions to the building code in sub-areas of the city[1] — these documents are the focus of this paper. In order to allow automated compliance checking of some aspects of a building submission, we consider methods for automatically converting the rules written as text in these documents into a formal representation to allow automated reasoning on these rules. In this paper, we describe the creation of a manually annotated corpus of sentences from these documents as well as experiments with rule-based and machine learning approaches to automatically extracting the rules.

The main contributions of this paper are the following:

- a formal rule representation for building regulations of the Zoning Map of the City of Vienna
- annotation guidelines for 100+ attributes, developed with domain experts
- the BRISE-Plandok corpus of 7,000 sentences from building regulations with gold standard rule annotation
- a rule-based system for extracting rules from text via semantic parsing, evaluated on the BRISE-Plandok corpus
- baseline supervised learning systems for the attribute extraction task, trained and evaluated on the BRISE-Plandok corpus

The paper is structured as follows. Section 2 presents related work on rule corpora and rule extraction. Section 3 describes the rule extraction task. Section 4 presents data preprocessing steps. Section 5 documents the annotation process. Section 6 presents the rule-based system for mapping text to formal rule representations, Section 7 evaluates the rule-based system and machine learning baselines. Section 8 concludes the paper. The BRISE-Plandok corpus and all software described in this paper is available on GitHub[2] under an MIT license.

[1] referred to as *Besondere Bestimmungen* or Special Provisions
[2] https://github.com/recski/brise-plandok

2 Related work

Our overview begins with recent work on the construction of annotated corpora of legal texts in Section 2.1, followed by a review of approaches to the automatic extraction of rules from text. Section 2.2 surveys natural language processing applications in the construction domain, including several recent attempts at rule extraction for automated compliance checking.

2.1 Rule corpora and rule extraction

Despite increasing interest in the automatic analysis of legal text and the growing number of legal corpora, very few datasets contain annotation that explicitly encodes the contents of legal text. Popular tasks in legal natural language processing include prior case retrieval (Al-Kofahi, Tyrrell, Vachher, & Jackson, 2001; Shao et al., 2020), judgement prediction (Martin, Quinn, Ruger, & Kim, 2004; Strickson & De La Iglesia, 2020), summarization (Kanapala, Pal, & Pamula, 2019; Moens, Uyttendaele, & Dumortier, 1999), and semantic segmentation (Kalamkar et al., 2022; Saravanan, Ravindran, & Raman, 2008). Datasets used to develop and evaluate approaches to these tasks typically only include text annotation relevant to the given task. German legal corpora have also been developed for such applications, these include datasets for legal information retrieval (Wrzalik & Krechel, 2021) and for the summarization of German court rulings (Glaser, Moser, & Matthes, 2021). Datasets that formalize the contents of legal text include a corpus of business constracts translated to logical form (Governatori, 2005) and the LEDGAR dataset (Tuggener, von Däniken, Peetz, & Cieliebak, 2020) containing automatically generated labelings of provisions in material contracts made available by the U.S. Securities and Exchange Commission (SEC). A small manually labeled corpus of 601 sentences from the German Civil Code addresses the problem of classifying legal norms by semantic type (Waltl, Bonczek, Scepankova, & Matthes, 2019), a subtask of rule extraction that is similar to the modality classification task described in this paper. An approach to extracting norms from business contracts is presented by Aires, Pinheiro, Lima, and Meneguzzi (2017), who also contribute manual annotation of 9864 norms in 92 contracts from a corpus of Australian contracts (Curtotti & McCreath, 2011).

The approach to rule extraction presented in this paper relies on a combination of syntactic and semantic parsing for attribute extraction and simple pattern-based detection of attribute roles and rule modality. An earlier version of our system is evaluated on a manually annotated dataset of 10 documents and 193 sentences in Recski, Lellmann, Kovács, and Hanbury (2021). A similar combined approach is taken by Dragoni, Villata, Rizzi, and Governatori (2016), who use syntactic constituency parsing, information from the lexical ontology WordNet (Miller, 1995), and pattern-based term extraction for identifying concepts in the Australian Telecommunications Consumer Protections Code, and annotate them with deontic labels. Their pipeline for identifying logical relationships between chunks of text also includes the Boxer framework

(Ahn et al., 2005) for constructing Discourse Representation Structures using a Combinatory Categorial Grammar (CCG) parser and CCG supertagging (Curran, Clark, & Bos, 2007). Another combination of constituency parsing and pattern matching is presented by Wyner and Peters (2011) and is applied to an excerpt from the U.S. Code of Federal Regulations describing some compliance rules of the Food and Drugs Administration (FDA). This system also relies on a lexical ontology, logical relations are identified based on thematic roles that are detected using VerbNet (Kipper, Korhonen, Ryant, & Palmer, 2008).

2.2 NLP in the construction domain

The BRISE-Plandok corpus presented in this paper contains annotated building regulations and has been developed as part of a project for digitalizing the compliance review process of the City of Vienna. Recent approaches to the modeling and extraction of regulations in the construction domain vary greatly both in their choice of semantic representation and their methods for mapping text to such representations. Kruiper et al. (2021) create the `ScotReg` corpus of Scottish building regulations, define a sequence labeling task that is a combination of shallow parsing (chunking) and semantic role labeling, assigning labels such as `Action` and `Object` to spans of text that are also syntactic constituents, and annotate 200 sentences using this representation to create the SPaR.txt dataset, which they use to train a standard deep learning architecture consisting of BERT embeddings, bidirectional Long Short-Term Memory (bi-LSTM) and Conditional Random Fields (CRFs). On the test portion of the dataset their models achieve precision, recall, and F1 scores around 80%. The system described in Zhang and El-Gohary (2015) is a pipeline of rule-based systems for extracting attribute-value pairs from text and for using these to construct logic rules. The extraction component uses a combination of part-of-speech (POS) tagging, constituency parsing, as well as gazetteers of terms conveying negation, modality, units of measurement, and comparative relations. The rule construction module consists of a set of patterns for mapping sets of such attributes to logical formulae, followed by rules for conflict resolution. The authors evaluate their system on only quantitative requirements from a chapter of the International Building Code (IBC). On the task of extracting individual logic clause elements (concepts and relations) they report overall precision and recall values of 98.2% and 99.1%, respectively. The integration of this pipeline into a unified framework for automated compliance checking (ACC) is described in Zhang and El-Gohary (2017). A framework for iteratively expanding the set of rules used by this system is developed by Xue and Zhang (2022) using a human-in-the-loop approach, similar to our process for creating the rule-based attribute extraction system presented in Section 6 of this paper. Another NLP pipeline that implements rules operating over POS tags, constituents (chunks), and gazetteers, is that of Guo, Onstein, and Rosa (2021), who also automatically map the extracted elements to the ifcOWL

ontology using both string-based matching and WordNet-based semantic similarity using the Wu-Palmer metric (Wu & Palmer, 1994). The case study they describe involves applying the system to automated compliance checking of a single building, we do not have knowledge of any additional evaluation of the approach. Finally, another recent approach to the automatic parsing of building regulations is that of Fuchs, Witbrock, Dimyadi, and Amor (2022), who train an encoder-decoder model using the Transformer architecture (Vaswani et al., 2017) in an attempt to perform end-to-end parsing of natural language sentences into logic formulae. Their experiments are based on a dataset created by Dimyadi, Fernando, Davies, and Amor (2020) and containing a sample of the New Zealand Building Code (NZBC) mapped to the XML-based LegalRuleML (LRML) format (Athan et al., 2013). The training data is created by extending this corpus using data augmentation techniques and a variety of out-of-domain semantic parsing datasets. Evaluating multiple Transformer-based baseline architectures the authors report F-scores around 40%, measured on sets of atomic elements of the LRML formulae to reward partial matches in incorrect formulas. Manual error analysis concludes that despite the low figures "reasonable structure was learnt" (Fuchs et al., 2022, p.8).

While the focus of our review is the automatic processing of natural language regulations, we note that approaches to the semantic modeling of building regulations have also been developed for manual formalization. This work includes the development of an object model for a section of the England and Wales Building Regulations (Malsane, Matthews, Lockley, Love, & Greenwood, 2015), the translation of the Korean Building Act into an executable format (Lee, Lee, Park, & Kim, 2016), and the manual mapping of UK building codes to a regulation ontology (Beach, Rezgui, Li, & Kasim, 2015).

3 Task definition

Construction in the City of Vienna is regulated by the city's Building Code (*Bauordnung*[3]), which contains general rules as well as exceptions and special regulations specific to a particular area (*Plangebiet*). Nearly 1,200 text documents contain area-specific regulations, and extraction of their contents into structured, machine-interpretable form is a key task in the digitalization of the construction approval process undertaken in the BRISE project. In this section we first present the formal rule representation, then show how the task of mapping raw text to such representations was divided into several subtasks, in order to simplify the annotation process for non-experts and to enable automatic rule extraction with rule-based and machine learning methods.

3.1 Rule representation

Some typical sentences from the area-specific regulations are presented in Figure 1. Following an initial review of several hundred similar sentences and of legislation that explicitly defines the possible scope of such area-specific rules,

[3] https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=LrW&Gesetzesnummer=20000006

we developed a generic representation of rule contents using attribute-value structures in the JSON format. Formal representations of the example sentences in Figure 1 are presented in Figure 2. The `modality` field can take one of three values and indicates whether the rule conveys `obligation`, `permission`, or `prohibition`. The content of each rule is represented as a list of attributes, e.g. in sentence `6833_21_0` the value of `Dachart` 'roof type' is `Flachdach` 'flat roof'. The role of each attribute in the rule structure is indicated by its `type`, which can take one of 4 values. The `content` type indicates attributes that are regulated by the rule, as opposed to attributes of the type `condition` that describe the circumstances under which the rule applies.

`7158_5_1` *In der Bayerngasse sind Vorkehrungen für die Pflanzung einer Baumreihe zu treffen.*
'In Bayerngasse, provisions must be made for the planting of a row of trees.'

`6833_21_0` *In dem mit BBS bezeichneten Bereich sind die Dächer als Flachdächer auszubilden.*
'In the area marked BBS, roofs must be constructed as flat roofs'

Fig. 1 Sample sentences from area-specific regulations. Sentence IDs represent the location of the sentence, using the format *documentID_ sectionID_ sentenceID* (see Section 4 for details).

```
{
  "id": "7158_5_1",
  "modality": "obligation",
  "attributes": [
    {
      "name": "VerkehrsflaecheID",
      "value": "Bayerngasse",
      "type": "condition"
    },
    {
      "name": "VorkehrungBepflanzung",
      "value": "die Pflanzung einer Baumreihe",
      "type": "content"
    }
  ]
}
```

```
{
  "id": "6833_21_0",
  "modality": "obligation",
  "attributes": [
    {
      "name": "Planzeichen",
      "value": "BBS",
      "type": "condition"
    },
    {
      "name": "Dachart",
      "value": "Flachdach",
      "type": "content"
    }
  ]
}
```

Fig. 2 Formal representations of the rules in Figure 1

Both of the example sentences in Figure 2 contain one attribute each of these two types. Some rules also contain exceptions, these are represented by attributes of the types `conditionException` and `contentException`. A `conditionException` restricts the conditions of a rule, such as in `7922_3_1` *Nicht als Fußweg ausgewiesene Straßenquerschnitte bis unter 8,0 m sind als Wohnstraßen auszugestalten* 'Road cross-sections below 8,0 m that are not designated as sidewalks shall be designed as residential streets.'. A `contentException` describes an exception to the content attribute, such as

the prohibition in `7481_40_0` *Es darf nur ein Fachmarkt, aber kein Einkaufszentrum für Lebens- und Genussmittel der Grundversorgung errichtet werden.* 'Only a specialty store may be built, but not a shopping center for basic foodstuffs'. This sentence is also an example of a sentence containing multiple instances of the same attribute (in this case, `Widmung` 'dedication'), with different values. The full representation of this sentence is shown in Figure 3. Attribute values may be Booleans (true/false), e.g. the content of the prohibition in `7651_15_1` *Die Errichtung von Wohngebäuden ist untersagt.* 'The construction of residential buildings is prohibited.' is the attribute `VerbotWohnung` 'apartment prohibition' with the value `True`. All other kinds of values are represented as strings, even if they contain a numerical component, e.g. the attribute `GehsteigbreiteMin` 'minimum sidewalk width' will have values such as `2,0 m`, which is the literal string in the sentence denoting 2.0 meters (German uses the comma as the decimal separator), and its internal structure is not represented in the JSON format.

```
{
  "id": "7481_40_0",
  "modality": "permission",
  "attributes": [
    {
      "name": "Widmung",
      "value": "Fachmarkt",
      "type": "content",
    },
    {
      "name": "Widmung",
      "value": "Einkaufszentrum für Lebens- und Genussmittel der Grundversorgung"
      "type": "contentException"
    }
  ]
}
```

Fig. 3 Formal representation of `7481_40_0` *Es darf nur ein Fachmarkt, aber kein Einkaufszentrum für Lebens- und Genussmittel der Grundversorgung errichtet werden.* "Only a specialty store may be built, but not a shopping center for basic foodstuffs". The rule contains two instances of the attribute `Widmung` 'dedication', with different types and values.

The generic rule representation introduced here has the advantage that it is independent of any particular formalism used for reasoning with the resulting rules. This is important because in the literature a plethora of different such frameworks have been introduced, using a number of different formal representations of the rules. Examples include dyadic deontic logic (considered here), defeasible deontic logic (Governatori, 2018), argumentation based approaches (Modgil & Prakken, 2014) or input output logic (Parent & van der Torre, 2013). Crucially, concepts like exceptions are modelled differently in the different frameworks, sometimes making it impossible to translate from one of these into another. In order to ensure usefulness of the obtained rules for as wide a variety of reasoning frameworks as possible, we use the intermediary representation described above. The slight disadvantage of this approach is, that the rules cannot be used "out of the box" in a standard reasoner, but this is outweighed by the flexibility gained by being able to translate them into

such a wide variety of formalisms. The rule format is based on some assumptions that hold for most but not all rules in the corpus, such as that there is always a one-to-one mapping between sentences and rules or that each rule has exactly one modality. Some consequences of these assumptions are mentioned in various sections of the paper as we document the corpus creation process, and a summary discussion of all such caveats is provided in Section 5.7.

3.2 Rule extraction

The rule format introduced in the previous section allows us to approach the task of automatic rule extraction in a step-by-step fashion. The pipeline of extraction steps described in this section is also the basis for the two-stage annotation process described in Section 5. A major source of difficulty in extracting rules from text is the diversity of topics discussed. Encoding all textual rules in the city's zoning plan required the definition of over 100 attributes, from `AnordnungGaertnerischeAusgestaltung` 'gardening design required' to `VorstehendeBauelementeAusladungMax` 'maximum protrusion of building elements' (the corresponding annotation guidelines of over 40 pages are available online[4] and are described in Section 5). We isolate this most challenging aspect of the rule extraction task by dividing it into two steps applied to each sentence:

1. detecting which of the defined attributes are mentioned, and
2. extracting the full rule representation.

The first step of the annotation process as well as of each of the rule-based and ML-based solutions we implement is the detection of attributes mentioned in each sentence, without regard to their values, types, or the modality of the rule conveyed by the sentence. This task, which we shall refer to as *attribute extraction*, is a multi-label classification task. Each sentence is mapped to the set of attributes they mention, without regard to their value or role in the full rule representation. Sentences containing multiple instances of the same attribute, such as the example in Figure 3, are labeled with a single instance. This greatly simplifies both the annotation process and the automated labeling, and multiplicity of labels can be reliably detected in the second step, when attribute types and values are determined. Sentences not containing any rules are mapped to the empty set, those conveying a rule typically contain up to 3 attributes, but some contain as many as 6 (detailed descriptive statistics about the annotated dataset are provided in Section 5.6).

Once sentences have been labeled with the attributes they mention, the full rule representation can be extracted by classifying attributes by type (`condition, content`, etc.), finding their values, and determining the modality of the sentence. Each of these pieces of information may either be indicated by the rule text, such as numerical and textual values of attributes or the modality of most rules, but in many cases they can also be inferred from the semantics of the attributes. For example, some Boolean attributes always

[4] https://github.com/recski/brise-plandok/tree/main/guidelines

appear with the same value (such as `AnordnungGaertnerischeAusgestaltung` 'gardening design required', which, if present, is always `True`), while others always have the same type (such as `DurchfahrtBreite` 'passage width', which only appears as `content`). The two-stage process of annotation described in Section 5 ensures that such inferences are made automatically before performing human annotation of types, values, and modalities.

4 Data preprocessing

The zoning plan of the City of Vienna and accompanying text documents are available online[5] to the general public. Text regulations are published in PDF documents, one for each planning area. The total number of documents is close to 1500, the exact number changes as new documents are issued or existing ones retracted. For the corpus presented in this paper we used documents downloaded in December 2020. PDF filenames indicate the four-digit ID number of the planning area they refer to (e.g. `Plandokument_7299.pdf`), the number may be followed by one or two additional characters indicating if the document is supplementary to the main document regulating a given area. 1433 PDF documents were downloaded, of which 256 contained image data only, these were not processed further. We then used the `pdftotext`[6] tool to extract the text content from the remaining 1177 documents.

The next steps involved the segmentation of text documents into sections, then into sentences and words. The `pdftotext` tool preserves the visual layout of text in PDF documents using newline characters in its output. This allowed us to infer the section structure of each document based on line breaks and using regular expressions for detecting section headers. Whitespace characters were then normalized in each section, any number of consecutive whitespace characters was replaced by a single space. Finally, the text in each section was split into sentences and words using the default German model of the `stanza`[7] NLP library. Many errors in the detection of sentence boundaries was caused by abbreviations such as *Abs.* (*Absatz* 'paragraph'), *Zl.* (*Zahl* 'number'), *Kat.* (*Kategorie* 'category'), etc. A second group of segmentation errors was caused by dates containing the name of a month after the number indicating the day, e.g. *1. Oktober* 'October 1st', which also caused stanza to falsely detect a sentence boundary. All of these errors were fixed by inserting an additional processing step into the stanza pipeline, implemented as the stanza processor `fix_ssplit`[8] in the `tuw-nlp` library, which reverses splits by merging pairs of consecutive sentences if the first one ends in one of 8 abbreviations that we listed manually[9], or if the split occurs before the name of a month. We also use this module to undo splits on colons (:), to ensure that all information relevant to a rule remains in a single sentence, such as in the sentence `7808_12_0` *Für*

[5] https://www.wien.gv.at/flaechenwidmung/public/
[6] https://poppler.freedesktop.org/
[7] https://stanfordnlp.github.io/stanza/
[8] https://github.com/recski/tuw-nlp/blob/main/tuw_nlp/text/segmentation.py
[9] The 8 abbreviations processed by the `fix_ssplit` module are the following: *Abs., Zl., Pr., Kat., Kat.G., lit., ONr., bzw.*

die mit BB2 bezeichnete Grundfläche wird bestimmt: Es dürfen keine Bauwerke errichtet werden. 'For the area marked by BB2 it is determined: no structures may be erected'. Sentences output by the `fix_ssplit` module are then processed further by default stanza components for tokenization, part-of-speech (POS) tagging, and universal dependency (UD) parsing. These analyses are used by the pattern-based rule extraction system described in Section 6, which supports the human annotation process (see Section 5) and is also evaluated as a standalone solution in Section 7.

5 Annotation and corpus creation

We document the annotation and review process implemented to create the BRISE-Plandok corpus. Section 5.1 provides a high-level overview, Section 5.2 describes the annotation guidelines, and Section 5.3 documents all technical details of the annotation process. Section 5.4 presents the review process used to finalize the gold standard annotation of the corpus, Section 5.5 describes postprocessing steps, Section 5.6 provides statistics about the annotation process (agreement and performance) and about the final BRISE-Plandok corpus, and Section 5.7 discusses some caveats.

5.1 Overview

In Section 3 we have defined the rule extraction task as a series of two subsequent steps. *Attribute extraction* is the multi-label classification task of establishing the set of attributes mentioned in a sentence, while *rule construction* includes the extraction of attribute values (strings, numbers, Boolean values, etc.), the classification of attributes by type (condition, content, etc.) and the classification of sentences by modality (obligation, permission, prohibition). The human annotation process was organized in two steps corresponding to these two tasks. All annotation output was subsequently reviewed by the authors to create the gold standard annotation of the BRISE-Plandok corpus. A high-level overview of the annotation and review process is presented in Figure 4, details of each step are documented in Section 5.3.

The attribute extraction task is the first and most complex task in the annotation process. Non-expert annotators are required to choose for each sentence one of over a hundred different attributes based on written guidelines (see Section 5.2) and limited amount of direct training. We reduce this workload in multiple ways. A rule-based attribute extraction system provides high-precision suggestions that annotators need only check for correctness. Since the rule-based system was developed to predict the most frequent attributes in the data (see Section 6), this step could provide suggestions for over 90% of sentences containing rules, and nearly 80% of all gold attributes was based on a correct automatic suggestion (see Section 5.6 for detailed figures).

Additionally, the expert review of annotations (see Section 5.4) was performed in parallel and gold standard annotations established by experts were used to pre-annotate sentences in new documents for which the ground truth

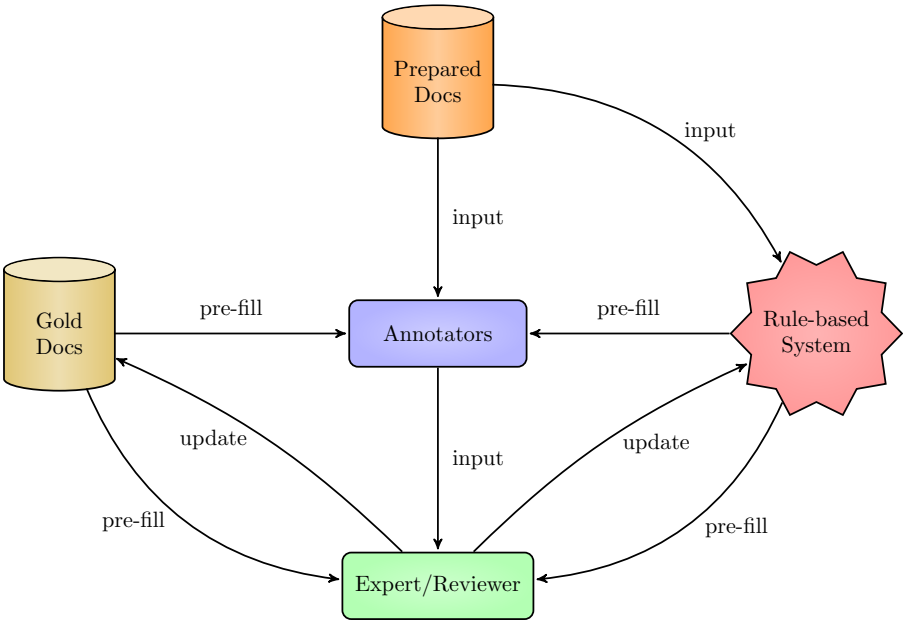


Fig. 4 High-level overview of the annotation and review process

annotation was already available from a previous document. The coverage of this step was increased by considering two sentences identical if they only differ in numerical characters, corresponding to the common case when different values of an attribute are specified in the same way. The decision to run the review process in parallel to the annotation by non-experts also allowed us to discover and resolve contradictions and ambiguities of the guidelines at an early stage.

5.2 Guidelines

The attribute extraction task is a multi-label annotation task that maps each sentence of the zoning plan documents to the set of all attributes mentioned in the text. The annotation guidelines defining each attribute were created in an iterative process. An initial set of required attributes was provided by domain experts working in the BRISE project to implement automated compliance checking of construction proposals. We constructed the first version of the guidelines using this list and the city's Building Code (*Wiener Bauordnung*[10], henceforth WBO) that regulates the types of requirements that zoning plan documents may contain. We then annotated an initial set of 5 documents using these guidelines, updating the definitions and adding examples to each attribute as we encountered them. We then organized a series of workshops

[10] https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=LrW&Gesetzesnummer=20000006

with 10 officers of the city's Building Authority (*MA37 Baupolizei*[11], henceforth MA37), representing all three regional offices (Vienna West, East, and South) of the Authority.

The officers of MA37 acted as domain experts to ensure the technical correctness of the attribute definitions and to identify ambiguities and missing attributes. After initial discussions that already lead to improvements of our guidelines, the 10 officers also performed the annotation task on small samples of documents. In the first phase, all experts annotated the same two documents, and later two batches of 4 documents each were independently annotated by groups of 3 or 4 annotators, with at least one officer representing each city region. We then calculated inter-annotator agreement for each attribute and each pair of annotators using Cohen's kappa, and ranked attributes based on average pairwise kappa across all annotators, to quickly identify problematic attributes. In each phase, our manual inspection of these attributes created the agenda for subsequent workshops that in turn led to corrections and simplifications of the annotation guidelines. Workshops were conducted separately for each region in the first phase and for each annotation group in the second phase. After the initial meetings in October 2020, 7 two-hour workshops were organized in the first phase and 13 workshops of 45 minutes each in the second phase. The last round of substantial updates and simplifications of the annotation guidelines was finished in April 2021 and the last workshop with MA37 took place in May 2021. All changes in the guidelines have been documented in its version history.

The final version of the annotation guidelines[12] defines 99 attributes, divided into 14 categories based on topic. This categorization is not in any way related to the formal task, it serves the convenience of the annotators navigating the document and the annotation spreadsheets (see Section 5.3 for details). The document is generated using custom Latex macros. In addition to the definitions it contains an index of attributes and a version history. Each attribute definition consists of a short description, the specification of the type of value the attribute takes (e.g. text, Boolean, integer or real number with or without a unit of measurement), and one or more example sentences from the zoning plan documents, where available, including sentence IDs for reference. Finally, each entry contains a citation of the specific passage of the WBO providing the legal basis for the use of the attribute in the zoning plan documents.

5.3 Implementation

We now give an overview of the human annotation process, followed by a detailed documentation of the tools and data formats used. The annotation of the BRISE-Plandok corpus was performed by 6 university students with German as a first language and without domain-specific expertise. Each annotator was contracted for 140 hours of work, to be performed in two phases between September and December of 2021, and paid by an hourly rate of 16,90 euros

[11] https://www.wien.gv.at/wohnen/baupolizei/
[12] https://github.com/recski/brise-plandok/tree/main/guidelines

for a total of €2,365. The first phase of the annotation included the attribute extraction task only. After an initial training phase using the same batch of 5 documents for each student, the remaining documents were annotated by two students each. In addition to periodic workshops for discussing their questions, students also had access to a Slack channel where they could ask questions and where our answers were visible to all students.

The first phase ended after one month, at which time annotators reported an average of 60 working hours per person and had annotated an average of 112 documents and an average of 3225 sentences. Our review of the annotated documents ran in parallel, and the tasks for the second phase of the campaign were defined based on our experiences. In the second phase students were asked to classify previously annotated attributes by type (content, condition, etc.) and to classify sentences by modality. Extracting values of attributes was not part of the annotation campaign, these fields were partly filled using regular expressions (see Section 6 for details) and manually checked during the review process. Remaining work hours were used to perform both phases of the annotation on additional documents. At the end of phase 2 a total of 400 documents and 11513 sentences have been annotated by two students each, of which we reviewed 250 documents with 7049 sentences. Remaining annotations may be used in future work for extending the BRISE-Plandok corpus, either by additional review or by converting annotations to silver standard labels via simple heuristics such as majority voting. A total of 65 hours were spent on the full review of the 250 gold standard documents. The data processing architecture used for the annotation and review process is represented in Figure 5.

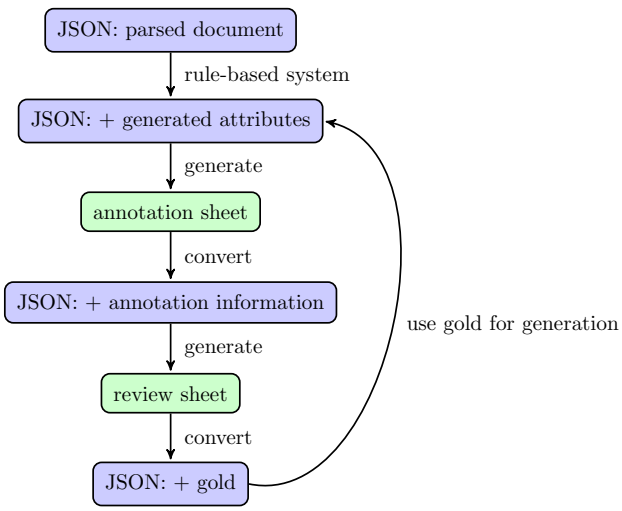


Fig. 5 Data evolution during annotation process.

We conducted a survey of free annotation tools before deciding to develop our own custom software. After reviewing the functionalities of doccano[13], Label Studio[14], brat[15], and INCEpTION[16], we determined that none of these libraries natively support the hierarchical labeling that is necessary to allow annotators to conveniently choose from a list of nearly a hundred labels for each sentence. We then developed a custom tool[17] based on the openpyxl[18] library that programatically generates `xlsx`-formatted spreadsheets from the JSON files containing the preprocessed documents. Both annotation and review was performed by editing such sheets locally, eliminating the need to deploy and maintain a web-based service.

Generated spreadsheets were distributed to annotators via shared cloud storage. The multi-label classification task was implemented using a row of pairs of dropdown menus for each sentence, one for choosing the attribute category and one for specifying the attribute, an example is shown in Figure 6. The coloring of cells was used to indicate pre-filled attributes suggested by the pattern-based system (see Section 6) and sentences for which attributes are present based on pre-existing gold standard annotation. These latter attributes were generated using a simple form of fuzzy matching that ignores digits, based on the observation that two sentences that differ only in numerical values always mention the same attributes. Annotators were instructed that pre-filled attributes in gray must be checked and may have to be corrected, while gold sentences need not be checked but may also be corrected.

ID	Sentence	Class1	
		Kategorie	Merkmal
8141_8_0	Bestimmungen ohne Bezeichnung des Geltungsbereichs mit dem Planzeichen BB:	Lage_Gelaende_Planzeichen	Planzeichen
8141_9_0	Der höchste Punkt der zur Errichtung gelangenden Dächer darf die ausgeführte Gebäudehöhe um höchstens 4,5 m überragen.	Dach	AbschlussDachMaxBezugGebaeude

Fig. 6 Annotation spreadsheet used in the first annotation phase. The yellow background indicates sentences with pre-existing gold annotations.

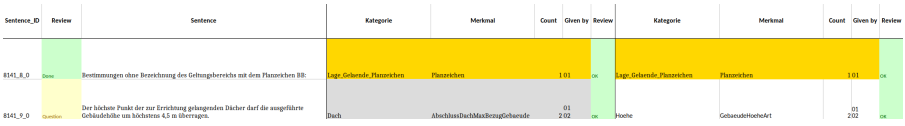


Fig. 7 Review spreadsheet used in the first annotation phase. The yellow background indicates sentences with pre-existing gold annotations, attributes with a gray background are pre-filled suggestions of the pattern-based attribute classifier.

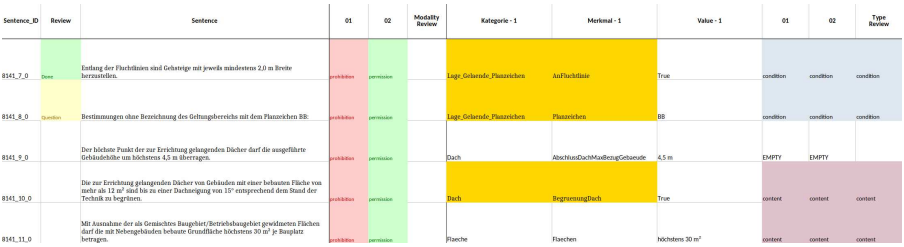
[13] https://github.com/doccano/doccano
[14] https://labelstud.io/
[15] https://brat.nlplab.org/index.html
[16] https://inception-project.github.io/
[17] https://github.com/recski/brise-plandok/tree/main/brise_plandok/annotation_process
[18] https://openpyxl.readthedocs.io/en/stable/

Completed spreadsheets were processed to extend the JSON representation of each document with fields containing the annotation provided by each student. These fields were then used in a subsequent step to generate the review spreadsheet, which presents a unified view of all annotations for each sentence and allows the reviewer to mark attributes as incorrect and/or to add missing attributes. Figure 7 shows an example of the review spreadsheet for the attribute extraction task. After review, this spreadsheet is processed to add the gold standard attribute labels to the JSON representations. Annotation spreadsheets were generated in batches so that sentences that have already been reviewed can be pre-annotated in the spreadsheets of new documents, greatly reducing the workload of annotators. Besides sentences that occur in many documents, this step also reduced the burden of scanning through boilerplate sentences, i.e. those that do not contain rules, since most of these are also reoccurring and thus became gold in the early stages of the parallel annotation and review process. When processing the completed spreadsheets, changes in the annotation of gold sentences triggered warnings for reviewers, this mechanism detects contradictions in reviewers' judgements and also the rare case of an annotator deciding to correct a gold attribute. For the second phase we generated spreadsheets that allow annotators to classify sentences by modality and to specify attribute types (condition, content, etc.). Figures 8 and 9 show sample spreadsheets for annotation and review, respectively.

Sentence_ID	Sentence	Modality	Kategorie	Merkmal	Value	Type
8141_7_0	Entlang der Fluchtlinien sind Gehsteige mit jeweils mindestens 2,0 m Breite herzustellen.	obligation	Lage_Gelaende_Planzeichen	AnFluchtlinie	True	condition
8141_8_0	Bestimmungen ohne Bezeichnung des Geltungsbereichs mit dem Planzeichen BB:	permission	Lage_Gelaende_Planzeichen	Planzeichen	BB	condition
8141_9_0	Der höchste Punkt der zur Errichtung gelangenden Dächer darf die ausgeführte Gebäudehöhe um höchstens 4,5 m überragen.	prohibition	Dach	AbschlussDachMaxBezugGebaeude	4,5 m	
8141_10_0	Die zur Errichtung gelangenden Dächer von Gebäuden mit einer bebauten Fläche von mehr als 12 m² sind bis zu einer Dachneigung von 15° entsprechend dem Stand der Technik zu begrünen.		Dach	BegruenungDach	True	content
8141_11_0	Mit Ausnahme der als Gemischtes Baugebiet/Betriebsbaugebiet gewidmeten Flächen darf die mit Nebengebäuden bebaute Grundfläche höchstens 30 m² je Bauplatz betragen.		Flaeche	Flaechen	höchstens 30 m²	content

Fig. 8 Annotation spreadsheet used in the second annotation phase. The yellow background indicates sentences with pre-existing gold annotations.

The zoning plan documents vary greatly in length, the average number of sentences in a document is 28.2 and the standard deviation is 14.5. When assigning documents to annotators, our goal was to distribute the workload equally, while also ensuring that each document is annotated by two annotators, but without fixed pairs of annotators. We therefore generated the 15 possible pairs of annotators and constructed 5 disjoint sets of 3 pairs, which we cycled through when distributing batches of documents. In order to partition each batch into balanced subsets of documents, an example of the NP-hard multiway number partitioning problem, we used an implementation of the largest differencing method (Karmarkar & Karp, 1982) from the



Sentence_ID	Review	Sentence	01	02	Modality Review	Kategorie - 1	Merkmal - 1	Value - 1	01	02	Type Review
8141_7_0	Done	Entlang der Fluchtlinien sind Gehwege mit jeweils mindestens 2,0 m Breite herzustellen.	prohibition	permission		Lage_Gekennde_Planzeichen	AnFluchtlinie	True	condition	condition	condition
8141_8_0	Question	Bestimmungen ohne Bezeichnung des Geltungsbereichs mit dem Planzeichen BB:	prohibition	permission		Lage_Gekennde_Planzeichen	Planzeichen	BB	condition	condition	condition
8141_9_0		Der höchste Punkt der zur Errichtung gelangenden Dächer darf die ausgeführte Gebäudehöhe um höchstens 4,5 m überragen.	prohibition	permission		Dach	AbschlussDachMaxBezugGebaeude	4,5 m	EMPTY	EMPTY	
8141_10_0		Die zur Errichtung gelangenden Dächer von Gebäuden mit einer bebauten Fläche von mehr als 12 m² sind bis zu einer Dachneigung von 15° entsprechend dem Stand der Technik zu begrünen.	prohibition	permission		Dach	BegruenungDach	True	content	content	content
8141_11_0		Mit Ausnahme der als Gemischtes Baugebiet/Betriebsbaugebiet gewidmeten Flächen darf die mit Nebengebäuden bebaute Grundfläche höchstens 30 m² je Bauplatz betragen.	prohibition	permission		Flaeche	Flaechen	höchstens 30 m²	content	content	content

Fig. 9 Review spreadsheet used in the second annotation phase. The yellow background indicates sentences with pre-existing gold annotations.

`numberpartitioning` [19] library. This process yielded a highly balanced distribution, each annotator was assigned at least 3457 and at most 3460 sentences. All custom software used in the process described in this section is available from the `brise-plandok` repository, together with detailed documentation [20].

5.4 Review

Due to the complexity of both the attribute extraction and rule construction tasks it was not possible to create a gold standard dataset automatically from labels assigned by non-expert annotators. As described in the previous section, each document was also subjected to manual review by the authors, for both annotation steps. The implementation details of the review process have been discussed in Section 5.3, in this section we document our experiences and the decisions we made during the review process that allowed us to create a consistent gold standard annotation of the BRISE-Plandok corpus.

The attribute extraction task that we defined in Section 3, and for which annotation guidelines were developed in an iterative process with domain experts (see Section 5.2), introduces a mapping of all topics addressed in the zoning plan documents to a set of 99 attribute labels. This taxonomy was designed to make the annotation task as clear as possible, nevertheless the full set of reviewed documents contained several examples of ambiguities that had to be resolved in a consistent way. When encountering topics that did not clearly correspond to one of the defined attributes, we generally preferred not to annotate them with similar but essentially incorrect labels, even if the non-expert annotators were in agreement about which existing attribute is the closest match. For example, consider the sentence `7142_10_0` *Auf der mit BB3 bezeichneten, als Bauland/Industriegebiet gewidmeten Fläche sind über die maximal zulässige Gebäudehöhe hinaus technisch notwendige Aufbauten bis zu einer Höhe von 26,0 m zulässig.* "In the area marked BB3, designated as building land/industrial area, technically necessary superstructures up to a height of 26.0 m are permissible in addition to the maximum permissible building height," while it makes a reference to the maximum building height, it was not annotated with the attribute `GebaeudeHoeheMaxAbsolut` 'maximum

[19] https://github.com/fuglede/numberpartitioning
[20] https://github.com/recski/brise-plandok/tree/main/brise_plandok/annotation_process

building height in absolute value', since the numerical value for this attribute is not present in the sentence.

In case more than one attribute is applicable to the same piece of information in a sentence, our general principle was to choose the most specific one only. For example, the sentence `7916_7_0` *Flachdächer, deren Fläche je Gebäude 100 m² übersteigt, sind im Ausmaß von mindestens 60 v. H. ihrer Fläche nach dem Stand der technischen Wissenschaften zu begrünen.* "Flat roofs whose area per building exceeds 100 m² must be greened to the extent of at least 60 % of their area in accordance with the state of the art in technical science." was annotated with the attribute `AnteilDachbegruenung` 'percentage of greening' but not with the less specific `BegruenungDach` 'roof to green'.

In addition to the issues concerning the attribute extraction task, some ambiguities were also encountered in the rule construction phase, most of them due to the assumption that each sentence contains a single rule expressing a single modality (obligation, permission, or prohibition). This assumption, while clearly untrue for a small fraction of the sentences in the BRISE-Plandok corpus, greatly simplified the annotation and review process. An example of a sentence containing multiple rules is `7719_12_0` *Auf der mit Spk/P BB3 gekennzeichnete Grundfläche ist die Errichtung von oberirdischen Bauten untersagt, die Errichtung von unterirdischen Bauten ist gestattet.* "The construction of above-ground structures is prohibited in the area marked Spk/P BB3, the construction of below-ground structures is permitted," which could not be annotated with a single modality, in such cases we kept the value that had been selected by the annotators. In some other cases, a single modality cannot be applied uniformly to all attributes, e.g. the sentence `7527_10_0` *Entlang der mit BB2 bezeichneten Baulinien ist die Errichtung von maximal 3,0 m hohen Lärmschutzeinrichtungen bzw. vollflächigen Einfriedungen zulässig.* "Along the building lines marked BB2, the construction of noise protection facilities or full-surface enclosures of a maximum height of 3.0 m is permitted." was labeled as `permission`, but could also be interpreted as an `obligation` for the design and the height of the enclosure.

Simple statistics suggest that the non-expert annotation process greatly reduced the amount of expert effort necessary for creating the gold standard annotation. At the start of the first annotation campaign we observed that the full annotation of a sentence by the expert authors required an average of approx. 2.5 minutes, while the first reviews of annotated documents only required 1.5 minutes per sentence. By the end of the review process the time spent on a single sentence was less than one minute on average, which included not only the finalization of attributes, attribute types, and sentence modalities, but also the semi-automatic extraction of attribute values and the ongoing improvement of the rule-based system used to support the annotation and review process. In the end, the 840 person-hours (approx. 5 person-months) of non-expert annotation were augmented with a total of 90 person-hours (approx. 0.5 person-month) of expert review to create the gold

standard annotation of the BRISE-Plandok corpus, which also included time spent on improving the task-specific patterns used by the rule extraction system described in Section 6. Detailed evaluation of inter-annotator agreement and annotator performance is presented in Section 5.6.

5.5 Postprocessing

The final step of the corpus construction process involved automated postprocessing of annotations based on feedback from experts in the BRISE project responsible for integrating the output of our rule extraction module into a system for automated compliance checking of digital building plans. Most changes described in this section involve splitting or merging of attributes using simple patterns, to improve the inter-operability of our rule extraction module and the expert system used in the BRISE project. On a case-by-case basis we decided to also implement some of these changes in the BRISE-Plandok corpus, thereby updating the gold standard annotation established by the review process. The remaining transformations, which we did not consider valuable for the published dataset, are implemented in a project-specific application that is used to serve API requests based on the annotated dataset (for the 250 gold documents) and the output of our rule extraction system (for all other documents). For the sake of completeness we document both types of changes. Tables 1, 2, 3 and 4 show the changes implemented in the BRISE-Plandok corpus. Additionally, the `PlangebietAllgemein` 'general plan area' attribute was renamed to `GesamtePlangebiet` 'entire plan area'. Changes not implemented in the BRISE-Plandok corpus involve converting the JSON structure to a format compatible with the expert system, adding metadata about the type of the attributes and removing units from numeric values.

Migration to the updated format was performed in a semi-automated way. Sentences affected by an attribute change were annotated by manually confirming or correcting the output of the rule-based system that was also used to provide attribute suggestions to annotators, but at every step the rule system was updated with patterns describing the newly created attributes. This way the migration process resulted not only in updated attributes in the gold standard annotation but also in an improved rule system that also covers the new attribute set. This semi-automatic migration process took a total of 21 hours of manual work, 12 of which was needed for the the most complicated attribute, `WidmungUndZweckbestimmung`, which occured in 736 unique sentences in the dataset. Scripts used for the migration process are available in our GitHub repository[21].

5.6 Statistics

In this section we present statistics on inter-annotator agreement, annotator performance (evaluated by comparing annotations to gold labels), and the

[21] https://github.com/recski/brise-plandok/blob/main/brise_plandok/full_attribute_extraction/migration/README.md

New attribute	Example
`BebauteFlaecheMax` 'maximum area to build on'	*Die bebaute Fläche darf insgesamt nicht mehr als 300 m² betragen.* 'The built-in area must not be larger than 300 m².'
`BebauteFlaecheMaxProzentual` 'maximum area to build on in percentage'	*Es dürfen höchstens 20 v.H. dieser Grundflächen überbaut werden.* 'At most 20% of these areas can be built on.'
`BebauteFlaecheMaxNebengebaeude` 'maximum area to build on for a side building'	*Die mit Nebengebäuden bebaute Gesamtfläche darf je Bauplatz 30 m² nicht überschreiten.* 'Pro building plot the built-in area of a side building must not be more than 30 m².'
`BebauteFlaecheMin` 'minimum area to build on'	*Die Dächer von Gebäuden mit einer bebauten Fläche von mehr als 12 m² sind bis zu einer Dachneigung von 15° entsprechend dem Stand der Technik zu begrünen.* 'The roofs of buildings with a built-in area of at least 12 m² and with a roof pitch of maximum 15° are to be greened with state-of-the-art technologies.'

Table 1 The attribute `Flaechen` 'area' was split into four new sub-attributes.

accuracy of automated suggestions presented to annotators, as well as descriptive statistics about the final BRISE-Plandok dataset. The figures presented here are a summary of the more detailed statistics that are available from our repository[22] together with the code used to calculate them.

Inter-annotator agreement on the task of attribute extraction is calculated for each attribute and for each pair of annotators using Cohen's kappa, calculated using the implementation of `scikit-learn`[23]. Figures for the 20 most frequent attributes are presented in Table 5. Notably, the two common labels `WidmungUndZweckbestimmung` and `Flaechen` that were later split due to the high variety of topics they cover (see Section 5.5) show considerably lower average agreement scores than most other frequent classes. These labels also led to low annotator performance, these figures are presented for the top 20 most frequent attributes in Table 6.

Inter-annotator agreement on the task of classifying attributes by type (`condition`, `content`, etc.) is presented for the 20 most frequent attributes in Table 7. While average agreement scores are generally in the 0.8–1.0 range, once again the two complex attributes `WidmungUndZweckbestimmung` and `Flaechen` show considerably lower scores than other frequent attributes. These attributes are split for the final dataset (see Section 5.5), and this also reduces the variability of attribute types. For example, the split of the

[22] https://github.com/recski/brise-plandok/tree/main/brise_plandok/stat/docs
[23] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.cohen_kappa_score.html

New attribute	Example
`Widmung`	*Der Raum darunter ist der Widmung gemischtes Baugebiet/Geschäftsviertel zugeordnet.*
'dedication'	'The space below has the dedication of building land/business quarter.'
`Nutzungsart`	*Die Gebäude sind der Nutzung als Kindergarten zuzuführen.*
'type of usage'	'The buildings are to be used as kindergartens.'
`BBallgemein`	*Die Struktureinheiten StrE1, StrE2, StrE3, StrE4 und StrE5 bilden ein Strukturgebiet.*
'general building regulation signs'	'The structure elements StrE1, StrE2, StrE3, StrE4 and StrE5 form a structure area.'

Table 2 The attribute `WidmungUndZweckbestimmung` 'dedication and purpose' was split into three new sub-attributes.

New attribute	Example
`GebaeudeHoeheMaxAbsolut`	*Die Gebäudehöhe darf höchstens 4,0 m betragen.*
'maximum building height in absolute value'	'The building must not be higher than 4,0 m.'
`GebaeudeHoeheMaxWN`	*Die Gebäudehöhe ist mit 74,0 m ü.W.N. begrenzt.*
'maximum building height wrt to the Viennese zero'	'The height of building is restricted to 74,0 m above the Viennese zero.'

Table 3 The attribute `GebaeudeHoeheMax` 'maximum building height' was split into two new sub-attributes.

umbrella category `Flaechen` 'surfaces' shown in Table 1 creates the attributes `BebauteFlaecheMin` 'minimum area built on' and `BebauteFlaecheMax` 'maximum area built on', the first of which always appears as a `condition` (e.g. in `7230e_7_0` *Auf den mit BB5 bezeichneten Baulandflächen sind Dächer von Nebengebäuden mit einer bebauten Fläche von mehr als 30 m² als Flachdächer auszuführen* 'On the area marked BB5, roofs of side buildings occupying more than 30 m^2 shall be constructed as flat roofs') while the second is always the `content` (e.g. in `6817e_16_0` *Die bebaute Fläche der zur Errichtung gelangenden Hauptgebäude darf jeweils höchstens 200 m² betragen.* 'The area occupied by the main building shall not occupy more than 200 m^2'). Another attribute exhibiting low annotator agreement scores on type classification is `GebauedeHoeheArt` 'building height type', which describes how other attributes describing building height are to be interpreted and the question of their status as `condition` or `content` of a rule is unclear (see Section 5.7).

New value	Example
`Fluchtlinie`	*Bei einer Straßenbreite ab 10,00 m sind entlang der Fluchtlinien Gehsteige mit mindestens 2,00 m Breite herzustellen.*
'alignment line'	'From a street width of 10,00 m along the alignment lines sidewalks must be constructed with a minimum width of 2,00 m.'
`Baulinie`	*Ebenso ist an allen Baulinien die Errichtung von Erkern, Baikonen und vorragenden Loggien untersagt.*
'building line'	'The construction of oriels, balconies and protruding loggias are forbidden as well.'
`Baufluchtlinie`	*Entlang der mit BB15 bezeichneten Baufluchtlinie ist die Herstellung von Fenstern von Aufenthaltsräumen von Wohnungen zur Verkehrsfläche unzulässig.*
'building alignment line'	'Along the building alignment line marked with BB15, the construction of windows of habitable rooms facing traffic areas are not allowed.'
`Straßenfluchtlinie`	*Entlang der mit BB7 bezeichneten Straßenfluchtlinie ist die Errichtung einer vollflächigen Lärmschutzwand bis zu einer Höhe von 3,5 m zulässig.*
'street alignment line'	'Along the street alignment line marked with BB7, the construction of a full-surface noise protection wall up to a height of 3,5 m is allowed.'
`Verkehrsfluchtlinie`	*Entlang der mit BB9 bezeichneten Verkehrsfluchtlinie ist die Errichtung einer vollflächigen Lärmschutzwand zulässig.*
'traffic alignment line'	'Along the traffic alignment line marked with BB9, the construction of a full-surface noise protection wall is allowed.'
`Grenzlinie`	*Entlang der Grenzlinie ist die Errichtung von Einfriedungen untersagt.*
'border line'	'Along the border line the construction of enclosures are prohibited.'
`Grenzfluchtlinie`	*An der mit BB7 bezeichneten Grenzfluchtlinie bzw. Verkehrsfluchtlinie ist die Errichtung einer durchscheinenden Lärmschutzwand bis zu einer Höhe von 3,0 m zulässig.*
'border alignment line'	On the area and border alignment line marked with BB7, the construction of a transparent noise protection wall up to a height of 3,0 m is allowed.'

Table 4 The attribute `Anfluchtlinie` 'on the alignment line' was migrated from Boolean values to textual ones describing the type of the alignment line.

Attr	Freq	Avg	Avg_w	1,2	1,3	1,4	1,5	1,6	2,3	2,4	2,5	2,6	3,4	3,5	3,6	4,5	4,6	5,6
Number of sentences				376	79	174	339	358	313	333	88	19	36	137	33	483	88	345
Planzeichen	1987	.98	.99	.98	.95	.96	1	1	.99	.99	.97	.98	.99	.94	1	1	.95	.98
WidmungUndZweckb.	1161	.82	.88	.95	.93	.92	.86	.97	.92	.92	.64	.68	.97	.82	1	.86	.00	.83
Flaechen	791	.88	.90	.94	.90	.79	.82	.99	1	.92	.84	.95	.95	.78	1	.95	.75	.66
VerkehrsflaecheID	457	.97	.98	1	1	1	1	1	1	1	.88	1	.94	.83	1	.99	.90	.98
AnFluchtlinie	393	.88	.93	.94	.84	.62	1	.95	1	1	.59	.89	.98	.94	1	.91	.65	.95
AnordnungGaertnerischeA.	356	.94	.95	.91	1	.95	.96	.97	.98	.92	.94	.90	.98	1	1	.91	.82	.93
GebaeudeHoeheMax	333	.93	.95	.96	.90	1	1	.97	1	.96	.79	.98	.93	.74	1	.92	.85	.96
Dachart	303	.97	.98	1	1	.92	.98	1	.95	1	1	1	1	.92	1	1	.81	.97
WidmungInMehrerenEbenen	294	.90	.90	.90	1	.92	.92	1	.82	.79	1	.88	1	.44	1	.92	1	.89
GebaeudeHoeheArt	290	.91	.95	.93	.66	.80	1	1	1	.97	.65	.87	1	1	1	.95	.82	.97
VorkehrungBepflanzung	282	.96	.98	.97	1	1	.98	1	.98	.97	.92	.95	1	.91	.98	1	.79	1
ErrichtungGebaeude	271	.70	.76	.95	.00	.50	.87	1	.83	.81	.00	1	.91	.65	1	.32	1	.66
PlangebietAllgemein	269	.79	.86	.58	1	.85	1	.96	1	1	-.03	.74	.79	1	1	.91	-.02	1
VonBebauungFreizuhalten	255	.85	.87	.92	.65	.82	.96	1	.88	.91	1	.94	.86	.49	1	.74	.79	.78
BegruenungDach	247	.98	.99	1	1	1	1	1	1	.96	1	1	.95	1	1	.97	.88	1
AbschlussDachMaxBezugG.	235	.97	.98	1	1	.85	1	1	1	.97	1	1	1	1	1	.98	.79	1
GehsteigbreiteMin	213	.98	.99	1	1	1	1	1	1	1	1	.94	1	1	1	.96	.85	1
StrassenbreiteMin	207	.95	.96	.85	1	1	1	1	1	1	.69	1	.91	.79	1	1	1	.93
GebaeudeBautyp	207	.97	.97	.98	.88	1	.96	1	1	1	1	1	1	.92	1	.93	1	.87
AufbautenZulaessig	176	.98	.98	1	1	1	1	1	1	1	1	.96	.96	1	.95	1	.79	.97

Table 5 Pairwise inter-annotator agreement on the task of attribute extraction for the 20 most frequent attributes in the dataset. `Avg` is the average of pairwise values, $\texttt{Avg}_w$ is weighted by the number of sentences annotated by each pair of annotators. `Freq` is the number of sentences in which the attribute appears either as gold or as annotated, and is used only to rank attributes in the table

Name	Freq	Precision	Recall
Planzeichen	1844	97.56%	97.50%
WidmungUndZweckbestimmung	916	69.87%	45.46%
VerkehrsflaecheID	400	90.87%	83.69%
AnordnungGaertnerischeAusgestaltung	292	86.65%	96.92%
Dachart	282	98.36%	96.68%
Flaechen	281	42.19%	97.39%
AnFluchtlinie	276	74.71%	91.10%
VorkehrungBepflanzung	274	98.49%	95.66%
GebaeudeHoeheArt	242	92.05%	93.73%
GebaeudeHoeheMax	232	80.59%	98.57%
BegruenungDach	220	96.61%	98.59%
WidmungInMehrerenEbenen	219	79.01%	91.86%
AbschlussDachMaxBezugGebaeude	219	95.67%	99.29%
ErrichtungGebaeude	213	78.58%	61.01%
GehsteigbreiteMin	207	100.00%	99.05%
PlangebietAllgemein	189	83.21%	68.96%
StrassenbreiteMin	177	92.62%	96.43%
GebaeudeBautyp	175	90.04%	98.57%
UnterbrechungGeschlosseneBauweise	154	98.28%	100.00%
AufbautenZulaessig	152	90.96%	97.47%

Table 6 Average annotator performance on the attribute extraction task for the 20 most frequent attributes in the dataset. `Freq` is the number of gold attributes in the dataset after review and before postprocessing.

These attributes also cause low annotator performance on the gold dataset, as shown in Table 8.

We also evaluate annotator agreement and performance on the task of classifying sentences by modality. The avarage pairwise agreement between annotators measured by Cohen's kappa is 0.737. Overall annotator performance (micro-average over all annotators) is presented in Table 9. We measure precision both on the full dataset and on the set of sentences containing rules, this shows that many false positives were caused by the fact that the modality classification task implicitly included the task of deciding whether a sentence contains a rule or not.

Finally, we evaluate the automated suggestions provided to the annotators for the task of attribute extraction. These suggestions were based not only on our rule-based attribute extraction system (described in Section 6 and evaluated on the final dataset in Section 7), but also on existing gold annotation for sentences that occur repeatedly in documents and which have already undergone review as part of another document (see Section 5.3 for details). In order to evaluate the contribution of this process to the manual annotation process we evaluate the correctness of suggested attributes at the time they were suggested to annotators. These figures are presented in Table 10, both for the full set of sentences and for those containing a rule. This once again illustrates

Attr	Freq	Avg	Avg_w	1,2	1,3	1,4	1,5	1,6	2,3	2,4	2,5	2,6	3,4	3,5	3,6	4,5	4,6	5,6
Planzeichen	1987	.93	.93	1	1	1	1	1	1	1	1	1	1	1	.00	1	1	1
WidmungUndZweckb.	1161	.44	.49	.78	.00	.00	.91	.57	.59	.75	.00	.29	.24	.30	.31	.41	1	.52
Flaechen	791	.46	.36	.93	1	.00	.64	.47	.58	.00	.73	1	.00	.60	.00	-.03	1	.00
VerkehrsflaecheID	457	.80	.86	1	.00	1	-.06	1	1	1	1	1	.00	1	1	1	1	1
AnFluchtlinie	393	.93	.90	1	1	1	1	1	1	1	1	1	.00	1	1	1	1	1
AnordnungGaertnerischeA.	356	.88	.87	.94	1	1	.68	.77	1	1	.77	.62	.83	1	.91	.69	1	1
GebaeudeHoeheMax	333	.72	.67	1	1	1	1	.76	.62	.64	1	.49	.74	.00	.44	.20	1	.90
Dachart	303	.86	.84	.89	1	.58	.89	.90	.66	.87	1	1	.82	1	.73	.80	1	.73
WidmungInMehrerenEbenen	294	.73	.36	1	1	.00	.00	1	1	1	1	1	.00	1	1	1	1	.00
GebaeudeHoeheArt	290	.34	.21	.00	-.25	1	.00	1	.00	.00	1	.00	.60	1	.00	-.22	1	.00
VorkehrungBepflanzung	282	.93	.89	1	1	1	1	1	1	1	1	1	1	1	1	.00	1	1
ErrichtungGebaeude	271	.93	.88	1	1	1	1	1	1	.00	1	1	1	1	1	1	1	1
PlangebietAllgemein	269	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
VonBebauungFreizuhalten	255	.87	.81	1	1	.00	1	1	1	1	1	1	1	1	1	1	1	.00
BegruenungDach	247	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
AbschlussDachMaxBezugG.	235	.66	.66	1	.00	1	1	1	1	-.08	1	.00	.00	1	1	-.04	1	1
GehsteigbreiteMin	213	.93	.90	1	1	1	1	1	1	1	1	1	-.08	1	1	1	1	1
StrassenbreiteMin	207	.93	.88	1	1	1	1	1	1	1	1	1	-.08	1	1	1	1	1
GebaeudeBautyp	207	.35	.43	.59	.50	.40	.62	1	.43	.26	.00	-.50	.18	.59	.19	.39	.00	.55
AufbautenZulaessig	176	.80	.73	1	1	1	1	1	1	1	1	.00	.00	1	1	1	1	.00

Table 7 Pairwise inter-annotator agreement on the task of classifying attributes by type, presented for the 20 most frequent attributes in the dataset. `Avg` is the average of pairwise values, $\texttt{Avg}_w$ is weighted by the number of attributes annotated by each pair of annotators. `Freq` is the number of sentences in which the attribute appears either as gold or as annotated, and is used only to rank attributes in the table

.

Attr	Freq	Precision	Recall	F1
Overall		92.50%	80.99%	86.36%
Planzeichen	2932	99.52%	98.09%	98.80%
WidmungUndZweckbestimmung	1520	74.67%	30.07%	42.87%
Flaechen	570	92.55%	91.58%	92.06%
VerkehrsflaecheID	322	96.13%	92.55%	94.30%
AnFluchtlinie	548	99.62%	95.07%	97.29%
AnordnungGaertnerischeAusgestaltung	584	95.20%	91.78%	93.46%
GebaeudeHoeheMax	430	86.70%	84.88%	85.78%
Dachart	454	93.36%	89.87%	91.58%
WidmungInMehrerenEbenen	116	86.32%	70.69%	77.73%
GebaeudeHoeheArt	484	46.81%	45.45%	46.12%
VorkehrungBepflanzung	534	99.80%	94.38%	97.02%
ErrichtungGebaeude	426	99.63%	63.38%	77.47%
PlangebietAllgemein	376	100.00%	81.12%	89.57%
VonBebauungFreizuhalten	288	98.88%	92.01%	95.32%
BegruenungDach	440	100.00%	98.64%	99.31%
AbschlussDachMaxBezugGebaeude	438	97.47%	96.58%	97.02%
GehsteigbreiteMin	310	99.34%	96.45%	97.87%
StrassenbreiteMin	354	99.13%	96.33%	97.71%
GebaeudeBautyp	344	72.46%	70.35%	71.39%
AufbautenZulaessig	304	98.63%	94.41%	96.47%

Table 8 Average annotator performance on the task of classifying attributes by type, overall and for the 20 most frequent attributes in the dataset. Figures are micro-averages over both type labels and annotators, more detailed figures for all attributes are available online (see recent footnotes for URLs). Sentences containing multiple instances of the same attribute are omitted to simplify evaluation.

Modality	Prec. (sens)	Prec. (rules)	Recall
prohibition	91.99%	94.00%	76.84%
permission	85.26%	88.47%	72.62%
obligation	85.32%	93.03%	96.30%
EMPTY	97.24%		89.10%

Table 9 Annotator performance on the task of classifying sentences by modality. Precision is calculated both on the full set of sentences and on sentences that contain rules.

the additional complexity caused by the implicit task of deciding whether a sentence conveys a rule or not.

We conclude this section by presenting simple descriptive statistics about the final BRISE-Plandok dataset in Table 11. For the purposes of evaluating rule extraction methods (see Section 7) we split the corpus into sections for training, validation, and testing, figures for these sections are also included in the table. More comprehensive statistics about the dataset such as the

	Precision	Recall	F1
all sentences	78.94%	78.69%	78.81%
all rules	84.44%	79.34%	81.81%

Table 10 Correctness of automatically suggested attributes at the time of annotation, evaluated on all sentences and on those containing a rule.

distribution of attributes, types, and modalities are available online[24]. The final dataset contains JSON entries for each document that retain complete information of the state of each sentence at every stage of the annotation and review process, including the pre-filled labels presented to annotators, labels provided by each annotator for each field, and the output of expert review before the postprocessing phase. Comprehensive documentation of all fields in the dataset is available online[25], their availability should enable additional analysis of the complex annotation and review process documented in this section.

	all	train	valid	test
Documents	250	200	25	25
Sentences	7049	5491	875	683
Sen. with segmentation error	215	123	74	18
Sen. with attributes	4238	3318	515	405
Rules	3994	3154	465	375
Median no. of rules per document	14	13	18	16
Attributes	9665	7714	1064	887
Median no. of attributes per rule	2	2	2	2

Table 11 Basic statistics describing the BRISE-Plandok corpus

5.7 Caveats

Finally we summarize known shortcomings of our rule representation, the annotation process, and the final dataset. When developing the rule representation described in Section 3, we take advantage of the fact that most sentences in the Zoning Plan contain at most a single rule, which can be represented as a set of attribute-value pairs, with their types (`condition`, `content`, etc.) indicating the role of each attribute, and which together with the sentence's modality allow us to construct a formal rule from the representation, e.g. using dyadic deontic logic. Sentences that contain multiple rules, such as `7719_12_0` *Auf der mit Spk/P BB3 gekennzeichnete Grundfläche ist die Errichtung von oberirdischen Bauten untersagt, die Errichtung von unterirdischen Bauten ist gestattet.* "The construction of above-ground structures is prohibited in the area marked

[24] https://github.com/recski/brise-plandok/tree/main/brise_plandok/stat/docs
[25] https://github.com/recski/brise-plandok/blob/main/DATA.md

Spk/P BB3, the construction of below-ground structures is permitted.", cannot be properly represented using our representation. Furthermore, our representation is ambiguous for rules containing multiple attributes of type `condition`, since these may be interpreted as being either in a conjunctive or disjunctive relation. For example, in the sentence `7199_10_0` *Flachdächer bis zu einer Dachneigung von fünf Grad sind entsprechend dem Stand der technischen Wissenschaften zu begrünen.* 'Flat roofs with a pitch not exceeding 5 degrees must be greened using state of the art technologies.' contains the attributes `Dachart` 'roof type' and `DachneigungMax` 'maximum roof pitch' that describe conditions that must both be met for the content attribute `BegruenungDach` 'roof to green' to apply. But in the sentence *Entlang der Haulerstraße und am Frankhplatz sind Vorkehrungen für die Erhaltung und Pflanzung von mindestens einer Baumreihe zu treffen.* 'Along Haulerstraße and on Frankhplatz, provisions shall be made for the preservation and planting of at least one row of trees.' the two `VerkehrsflaecheID` 'traffic area ID' attributes of type condition describe two locations for which the rule holds independently. Finally, a small number of rules in the dataset span over multiple sentences, these are also not captured by our representation.

6 Pattern-based rule extraction

In this section we present a rule-based solution to the task of mapping sentences to rule representations, as defined in Section 3. The semantic parsing system `text_to_4lang` (Kovács, Gémes, Kornai, & Recski, 2022; Recski, 2018) is used to map sentences of the building regulation documents to graph-based meaning representations, which are then used as input to multiple pattern-based sentence classifiers implemented via the POTATO library (Kovács, Gémes, Iklódi, & Recski, 2022) to extract attributes, and custom Python modules in the `brise-plandok` library for detecting attribute types, values, and rule modalities. An early prototype of this system with limited coverage was presented in (Recski et al., 2021), the improved solution described in this section achieves high accuracy across the entire BRISE-Plandok dataset (see Section 7 for detailed evaluation). As described in Section 5, components of this system were also used in the corpus annotation process to provide automatic suggestions, significantly reducing the workload of human annotators.

6.1 Semantic parsing

Our rule-based solution for attribute extraction operates over semantic graphs representing the meaning of each input sentence. Such graphs are constructed using the `text_to_4lang` algorithm, originally presented in (Recski, 2018) and improved in (Kovács et al., 2022). This method relies on Universal Dependency (de Marneffe, Manning, Nivre, & Zeman, 2021) parsing for detecting the syntactic structure of sentences (see Section 4), then maps each dependency parse tree to a directed graph of concepts following the `4lang` system of meaning representation (Kornai, 2010; Kornai et al., 2015). Figure 11 shows an example of

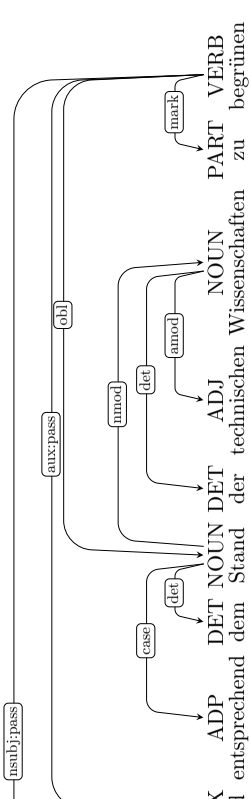


Fig. 10 Universal Dependency analysis of the sentence `7199_10_0` *Flachdächer bis zu einer Dachneigung von fünf Grad sind entsprechend dem Stand der technischen Wissenschaften zu begrünen.* 'Flat roofs with a pitch not exceeding 5 degrees must be greened using state of the art technologies.'

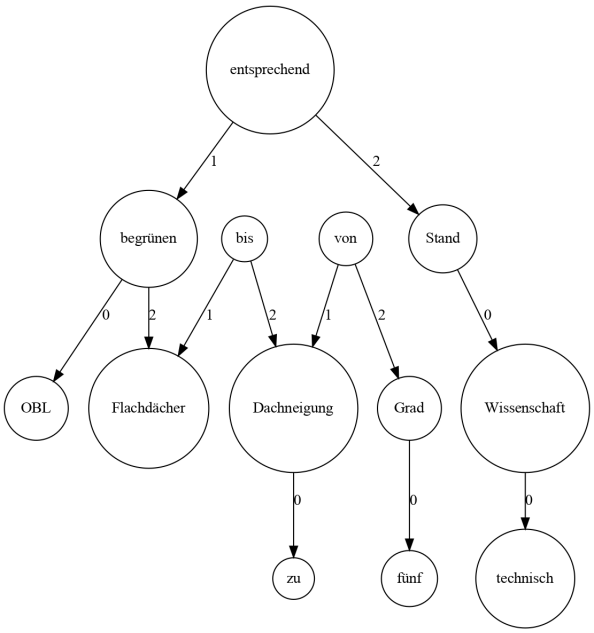


Fig. 11 4lang semantic representation of the sentence `7199_10_0` *Flachdächer bis zu einer Dachneigung von fünf Grad sind entsprechend dem Stand der technischen Wissenschaften zu begrünen.* 'Flat roofs with a pitch not exceeding 5 degrees must be greened using state of the art technologies.'

a `4lang` graph, built from the UD parse in Figure 10, representing the sentence *Flachdächer bis zu einer Dachneigung von fünf Grad sind entsprechend dem Stand der technischen Wissenschaften zu begrünen.* 'Flat roofs with a pitch not exceeding 5 degrees must be greened using state of the art technologies.' `4lang` graphs use three types of directed edges to connect words of an utterance. 1- and 2-edges connect binary relations to their arguments, while the 0-edge represents attribution, unary predication, and hypernymy. For a more detailed overview of the `4lang` formalism and the `text_to_4lang` system the reader is referred to (Kovács et al., 2022). For constructing these graphs we rely on the `text_to_4lang` implementation in the `tuw-nlp`[26] library. The patterns over UD parse trees used to construct `4lang` graphs are generally language-agnostic, a small number of rules have been added to handle German legal text. These rules directly map the modal auxiliaries *dürfen* 'may' *müssen* 'must' as well as the adjective *zulässig* 'permitted' and the verb *untersagen* 'forbid' to the `4lang` nodes `PER`, `OBL`, and `FOR`. Common words expressing negation (*nicht*, *kein*) are mapped to the `4lang` node `NEG`.

[26] https://github.com/recski/tuw-nlp

6.2 Attribute extraction

The semantic graphs constructed using the `text_to_4lang` system are used as input to a rule-based system for detecting mentions of attributes. Binary classifiers are built for 40 common attributes that together cover over 63% of all attribute mentions in the BRISE-Plandok dataset. A classifier for a single attribute is a list of graph patterns that are each matched against an input graph. If any pattern associated with an attribute is present in the graph, the attribute is predicted to be present in the input sentence. Each pattern is a directed graph, whose node and edge labels may be strings or regular expressions. For example, the graph pattern for the attribute `GebäudeHöheMax` 'maximum building height' in Figure 12 matches both of the graphs in Figures 13 and 14. Pattern-based classification is implemented using the `POTATO`[27] library (Kovács et al., 2022), the rule systems for each attribute are part of the `brise-plandok` repository. Patterns for each rule set have been collected to ensure high precision at the cost of lower recall. The overall precision of the complete system of 40 rule sets is above 93%. Detailed evaluation of the rule systems on the final dataset is presented in Section 7.

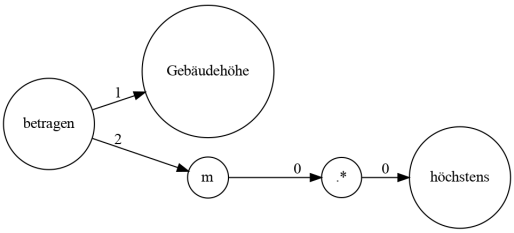


Fig. 12 Graph pattern for the attribute `GebäudeHöheMax` 'maximum building height'. *.** is the regular expression for matching any string

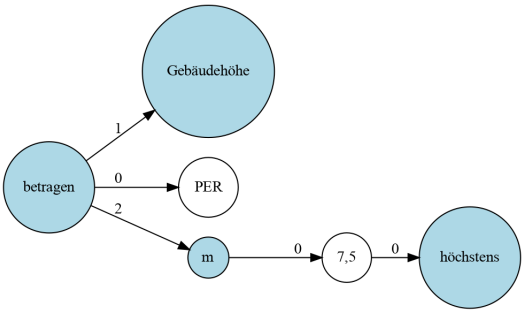


Fig. 13 4lang graph of the sentence `7870e_3_1` *Die Gebäudehöhe darf höchstens 16,0 m betragen.* 'The building height may not exceed 16.0 m.' Highlighted nodes indicate the subgraph matching the pattern in Figure 12

[27] https://github.com/adaamko/POTATO

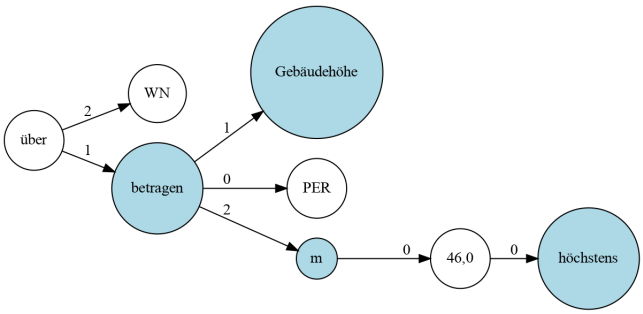


Fig. 14 4lang graph of the sentence `8250_56_0` *Die Gebäudehöhe darf höchstens +46,0 m über WN betragen.* 'The building height may not exceed +46.0 m above WN'. *WN* stands for *Wiener Null* 'Vienna Zero', the reference height for the City of Vienna. Highlighted nodes indicate the subgraph matching the pattern in Figure 12

6.3 Rule construction

Automatic construction of the complete rule representation introduced in Section 3.1 requires the classification of extracted attributes by type (condition, content, etc.), the extraction of their values from text, and the classification of each sentence by modality (obligation, permission, prohibition)[28]. The task of type detection, i.e. the mapping of each attribute to one of the four classes `content`, `condition`, `contentException` and `conditionException`, is greatly simplified by the fact that many attributes only or mostly occur with a single type. For example, `AnFluchtlinie` 'on the alignment line' is always a `condition`, and `AusnahmeGaertnerischAuszugestaltende` 'exception to the ordinance of horticultural design' is always a `conditionException`. Remaining cases are handled by custom patterns using regular expressions, an example is presented in Figure 15.

```
AttributesNames.DachneigungMax: {
    r"bis zu einer (Dachn|N)eigung von": {
        TYPE: AttributeTypes.CONDITION,
    },
    r"mit einer (Dachn|N)eigung bis": {
        TYPE: AttributeTypes.CONDITION,
    },
    ALL: {
        TYPE: AttributeTypes.CONTENT
    }
},
```

```
AttributesNames.DachneigungMax: {
    r"up to a roof pitch of": {
        TYPE: AttributeTypes.CONDITION,
    },
    r"with a roof pitch up to": {
        TYPE: AttributeTypes.CONDITION,
    },
    ALL: {
        TYPE: AttributeTypes.CONTENT
    }
},
```

Fig. 15 Type patterns for attribute `DachneigungMax` 'maximum roof pitch'. Two patterns define cases of `condition`, in all other cases we fall back to `content`.

Some Boolean attributes always have the same value by definition, e.g. `AnordnungGaertnerischeAusgestaltung` 'ordinance of horticultural design' is always True. Other Boolean values are detected using regular expressions

[28] The modality classification task relies on the assumption that each sentence contains at most one rule, see Section 5.7 for details

over the sentence, an example is shown in Figure 16. Finally, string and numerical values of attributes are detected using regular expressions with grouping and subpatterns that extract substrings from the input sentence, an example is presented in Figure 17. This limits the extraction to a sequence labeling task, rule representations are populated with attribute values as they are present in the input text and without additional normalization. All patterns used for type and value extraction are available on GitHub[29][30].

```
AttributesNames.AufbautenZulaessig: {
    r"nicht zulässig": {
        VALUE: False,
    },
    r"((?!nicht).)* zulässig": {
        VALUE: True,
    },
},
```

```
AttributesNames.AufbautenZulaessig: {
    r"not permitted": {
        VALUE: False,
    },
    r"((?!not).)* permitted": {
        VALUE: True,
    },
},
```

Fig. 16 Value patterns for Boolean attribute `AufbautenZulaessig` 'superstructure permitted'.

```
AttributesNames.GebaeudeHoeheMaxWN: {
    NUMBER_WITH_METER +
    r" über Wr. Null": {
        GROUP: 1,
    },
    NUMBER_WITH_METER + r" ü.W.N.": {
        GROUP: 1,
    },
},
```

```
AttributesNames.GebaeudeHoeheMaxWN: {
    NUMBER_WITH_METER +
    r" above the Viennese zero": {
        GROUP: 1,
    },
    NUMBER_WITH_METER + r" a.V.z.": {
        GROUP: 1,
    },
},
```

Fig. 17 Value patterns for attribute `GebaeudeHoeheMaxWN` 'maximum building height wrt to the Viennese zero'. Note that the "NUMBER_WITH_METER" variable is also a regular expression pattern. For this attribute in both cases the value is what is inside the 1st matching group.

The final step of rule construction is to determine the modality of each sentence (under the assumption that each sentence contains at most one rule, see Section 5.7 for a discussion), a three-way choice between the categories `obligation`, `permission`, and `prohibition`. Since nearly 75% of all sentences describes an `obligation`, we did not implement any classification algorithm during the annotation and review process. After the review was completed, we implemented a simple strategy for guessing the modalities of other sentences in the Zoning Plan based on the list of attributes found in each. For this we compiled two lists of attributes based on the gold standard data, those that most often occur in sentences with `permission` and `prohibition` modalities, respectively. The presence of these attributes[31] in a sentence will trigger the assignment of the respective modality, all other sentences will be classified as

[29] https://github.com/recski/brise-plandok/blob/main/brise_plandok/full_attribute_extraction/value/value_patterns.py

[30] https://github.com/recski/brise-plandok/blob/main/brise_plandok/full_attribute_extraction/type/type_patterns.py

[31] for the complete lists of attributes triggering each label, see https://github.com/recski/brise-plandok/blob/main/brise_plandok/full_attribute_extraction/modality/predict_modalities.py

`obligation`. This strategy achieves an accuracy of over 90% on the annotated dataset (see Section 7 for details).

7 Evaluation

In this section we present quantitative evaluation of our rule-based approach to the attribute extraction and rule construction tasks. The attribute extraction method is compared with a range of machine learning baselines, which are described in Section 7.1. Following standard practices we split the BRISE-Plandok dataset into three parts, designated `train`, `valid`, and `test`. The `train` portion is used to train ML models, `valid` is used for initial evaluation and optimization of hyperparameters, `test` is used for evaluation of our final models (the size of each of these portions is presented in Table 11). Sentences marked as involving a segmentation error (3% of all sentences, see Section 5.6) are disregarded during all evaluation. Results are presented in Section 7.2.

7.1 Machine learning baselines

We train and evaluate multiple supervised ML methods on the attribute extraction task to serve as baselines for comparison with our rule-based approach. Besides standard architectures we also include simple interpretable methods such as decision trees and decision lists, to illustrate their potential for learning high-accuracy symbolic models for text classification and to motivate further research on rule learning for information extraction. All models described below, except for the BERT-based method, are trained using bag-of-words features, with each word mapped to its lemma, as generated using our stanza-based NLP pipeline (see Section 4), and each sentence represented using a `CountVectorizer`[32], with `max_features`=3000, `max_df`=0.8 and `min_df`=0.001, these values were determined empirically using the validation set.

Logistic regression

A logistic regression model is trained using the implementation[33] of `scikit-learn`, with default parameters.

Decision tree

Decision tree classifiers are trained using the implementation[34] of `scikit-learn`, with maximum depth set to 5. Visualizations of the trees for each class are available in our repository[35].

[32] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

[33] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[34] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[35] https://github.com/recski/brise-plandok/tree/main/brise_plandok/baselines/output/decision_tree

Greedy Rule List

Using the `imodels` library we train a greedy rule list classifier[36] with a maximum depth of 5. This method learns Decision Lists (Rivest, 1987) by greedily splitting the training data based on one feature at a time based on gini value, but unlike decision trees creates only a single path of binary decisions. The learned lists for each attribute are available online[37].

One Rule

The One Rule classifier[38] (Holte, 1993) further restricts a Decision List to refer to only a single input feature. Combined with our bag-of-words representation, this model will determine a single lemma for each class such that the class label is predicted if and only if that word is present in the sentence. The learned rules for each attribute are available online[39].

BERT

A standard BERT-based classifier is trained using the model `bert-base-german-cased`[40] with default hyperparameters. Batch size was set to 64 and maximum token length was reduced to 282 to avoid out-of-memory errors. We used the `AdamW` optimizer[41] and a learning rate of 10^{-3}. Training was performed on a single GeForce GTX 1080 Ti GPU with 11 GB of RAM and ran for 300 epochs, evaluation was performed using the model state after 200 epochs, this was determined by selecting the model with the highest performance on the validation dataset. We also experimented with training separate classifiers for each attribute, but early experiments showed no significant performance gain.

7.2 Results

The quantitative performance of the rule-based attribute extraction system introduced in Section 6 is evaluated on the full BRISE-Plandok dataset and presented in Table 12. The machine learning models described in Section 7.1 are trained and evaluated both on the full set of 93 attributes present in the dataset and on the set of 40 attributes for which rule systems have been built, with the exception of the BERT baseline, which was trained on the smaller label set only. Overall results on the validation set are presented in Tables 13 and 14, results on the test set are in Tables 15 and 16. As expected, greedy rule learning approaches perform worse than our manually built rule systems, while more complex ML models perform comparably (BERT) or better (decision trees, logistic regression).

[36] https://csinva.io/imodels/rule_list/greedy_rule_list.html
[37] https://github.com/recski/brise-plandok/blob/main/brise_plandok/baselines/output/greedy_rule_list/REPORT.md
[38] https://csinva.io/imodels/rule_list/one_r.html
[39] https://github.com/recski/brise-plandok/blob/main/brise_plandok/baselines/output/one_rule/REPORT.md
[40] https://huggingface.co/bert-base-german-cased
[41] https://huggingface.co/docs/transformers/main_classes/optimizer_schedules#transformers.AdamW

	gold	pred	precision	recall	F1
overall (40 classes)	7383	6595	93.13	83.19	87.88
overall (93 classes)	9665			63.55	75.55
Planzeichen	1844	1855	94.82	95.39	95.11
Widmung	641	745	80.40	93.45	86.44
AnordnungGaertnerischeAusgestaltung	292	273	97.44	91.10	94.16
Dachart	282	266	99.25	93.62	96.35
AnFluchtlinie	276	261	96.17	90.94	93.48
VorkehrungBepflanzung	274	222	100.00	81.02	89.52
GebaeudeHoeheArt	242	210	94.29	81.82	87.61
BegruenungDach	220	211	96.68	92.73	94.66
AbschlussDachMaxBezugGebaeude	219	191	98.43	85.84	91.71
WidmungInMehrerenEbenen	219	107	82.24	40.18	53.99
BBAllgemein	217	228	87.28	91.71	89.44
GehsteigbreiteMin	207	168	100.00	81.16	89.60
GebaeudeHoeheMaxAbsolut	200	164	87.20	71.50	78.57
GebaeudeBautyp	175	167	95.81	91.43	93.57
Nutzungsart	157	117	79.49	59.24	67.88
UnterbrechungGeschlosseneBauweise	154	154	100.00	100.00	100.00
AufbautenZulaessig	152	132	98.48	85.53	91.55
VonBebauungFreizuhalten	144	58	86.21	34.72	49.50
DachneigungMax	132	45	100.00	34.09	50.85
DurchgangBreite	109	94	100.00	86.24	92.61
BebauteFlaecheMaxProzentual	107	84	95.24	74.77	83.77
AusnahmeGaertnerischAuszugestaltende	103	55	100.00	53.40	69.62
BauweiseID	97	56	98.21	56.70	71.90
DurchgangHoehe	96	90	93.33	87.50	90.32
VolumenUndUmbaubarerRaum	91	54	100.00	59.34	74.48
Stockwerk	80	63	93.65	73.75	82.52
DachflaecheMin	71	48	100.00	67.61	80.67
BebauteFlaecheMax	69	57	98.25	81.16	88.89
VerbotFensterZuOeffentlichenVerkehrsflaechen	67	57	100.00	85.07	91.94
BebauteFlaecheMaxNebengebaeude	65	71	81.69	89.23	85.29
VerbotWohnung	61	50	100.00	81.97	90.09
AnOeffentlichenVerkehrsflaechen	54	25	100.00	46.30	63.29
OeffentlicheVerkehrsflaecheBreiteMin	51	27	100.00	52.94	69.23
AnordnungGaertnerischeAusgestaltungProzentual	50	35	94.29	66.00	77.65
BebauteFlaecheMin	48	55	76.36	87.50	81.55
AnteilDachbegruenung	36	24	91.67	61.11	73.33
GebaeudeHoeheMaxWN	32	29	82.76	75.00	78.69
FBOKMinimumWohnungen	20	24	70.83	85.00	77.27
StellplatzregulativUmfangMinimumRelativ	18	19	94.74	100.00	97.30
StellplatzregulativUmfangMaximumRelativ	11	4	75.00	27.27	40.00

Table 12 Performance of the rule-based system on the full BRISE-Plandok dataset. Overall recall and F1-score is calculated both for the full set of attributes (93 classes) and for the 40 classes for which rules are provided. Attributes are listed in order of frequency in the dataset (`gold`), the `pred` column shows the number of attributes predicted by the rule-based system.

	Macro-avg			Micro-avg		
	P	R	F1	P	R	F1
Logistic regression	90.02	71.29	73.67	93.25	83.95	88.36
Decision tree	90.85	78.56	80.94	92.77	82.81	87.51
Greedy rule list	86.13	61.94	59.22	85.68	72.74	78.69
One rule	85.93	51.02	54.13	87.89	59.26	70.79

Table 13 Performance of ML baselines trained on all 93 classes, evaluated on the validation set (`valid`).

	Macro-avg			Micro-avg		
	P	R	F1	P	R	F1
Logistic regression	93.07	84.02	86.40	94.90	88.89	91.80
Decision tree	93.56	85.55	87.72	93.39	86.14	89.62
BERT	89.61	74.08	78.51	90.42	80.05	84.92
Greedy rule list	83.54	62.52	59.27	86.89	74.43	80.18
One rule	81.22	52.21	55.62	88.10	61.05	72.12
Rules	92.45	71.30	76.23	92.48	80.76	86.22

Table 14 Performance of ML baselines trained on the 40 classes handled by the rule-based system, evaluated on the validation set (`valid`).

	Macro-avg			Micro-avg		
	P	R	F1	P	R	F1
Logistic regression	91.59	76.01	77.10	93.91	86.20	89.89
Decision tree	89.59	82.45	82.59	92.68	83.69	87.96
Greedy rule list	84.09	61.57	58.48	84.77	72.98	78.43
One rule	84.46	51.46	53.50	87.38	62.37	72.79

Table 15 Performance of ML baselines trained on all 93 classes, evaluated on the `test` set.

	Macro-avg			Micro-avg		
	P	R	F1	P	R	F1
Logistic regression	93.84	76.37	79.99	95.26	88.75	91.89
Decision tree	92.38	85.31	86.45	93.97	86.61	90.14
BERT	90.87	73.77	77.03	91.12	83.33	87.05
Greedy rule list	85.72	57.67	58.67	88.36	74.64	80.93
One rule	84.66	49.98	53.56	90.18	65.38	75.81
Rules	93.97	74.14	78.19	93.54	82.48	87.66

Table 16 Performance of ML baselines trained on the 40 classes handled by the rule-based system, evaluated on the `test` set.

Manual inspection of the fully interpretable models such as the lemmas learned by the One Rule classifier[42] illustrates the inherent simplicity of the attribute extraction task for some frequent labels. For example, the attribute *AnordnungGaertnerischeAusgestaltung* 'gardening design required' is predicted with a precision and recall of .91 and 1.0, respectively, based on the presence of the lemma *gärtnerisch* 'related to gardening'. Unlike manually built patterns, however, such rules are prone to modeling the artefacts of the dataset rather than the semantics of attributes. For example, the attribute

[42] https://github.com/recski/brise-plandok/blob/main/brise_plandok/baselines/output/one_rule/REPORT.md

`AnFluchtlinie` 'along the alignment line' is predicted by the One Rule classifier based on the lemma *entlang* 'along', achieving .91 precision and .81 recall, comparable to the performance of the manually built 4lang pattern *(an|entlang)* $\xrightarrow{2}$ *.+linie*. While the quantitative performance of both systems remains below those of the machine learning architectures considered to be the state of the art in text classification, such interpretable models are often preferred in real-world applications due to their transparency and predictability. Detailed qualitative comparison of such models, including an analysis of the tradeoff between performance and interpretability, shall be presented in a forthcoming publication.

The detection of attribute types and the extraction of attribute values are evaluated on all gold attributes of the full BRISE-Plandok dataset in Table 17. The attribute-based modality prediction is also evaluated on the full dataset and compared to the baseline strategy of always choosing the most common modality `obligation`, these figures are shown in Table 18. Note that these rule-based systems were built manually based on insights from the full annotated dataset, therefore their ability to generalize across all documents of the Zoning Plan cannot be evaluated and their performance on unseen data may be lower.

	gold	predicted	precision	recall	F1
type classification	9673	9163	96.34	91.26	93.74
value extraction	9673	8679	90.00	80.75	85.12

Table 17 Performance of type classification and value extraction, evaluated on all gold attributes of the full dataset.

	Always obligation			Attribute-based		
	P	R	F1	P	R	F1
overall	74.69	74.69	74.69	90.64	88.03	89.32
obligation	74.69	100.00	85.51	93.50	95.51	94.49
permission	N/A	0.00	0.00	83.56	75.61	79.38
prohibition	N/A	0.00	0.00	74.43	53.12	61.99

Table 18 Performance of attribute-based modality prediction on the full dataset, compared to the baseline of always choosing the most common modality.

Finally, we evaluate the overall accuracy of rule construction on gold standard attributes, i.e. the ratio of sentences for which all attribute types and values as well as the rule modality was correctly detected. Table 19 presents these figures in three settings. The *gold attrs* setup uses ground truth attribute sets and only evaluates the automatic prediction of attribute values, types, and sentence modalities. Under these conditions two-thirds of all generated rules are fully correct. The *pred attrs (40)* setup uses attributes extracted by

our rule-based system, but discards attributes not among the 40 covered by the rules. This setup results in 55% fully correct rules, with attribute extraction being the dominant source of errors. Finally, *pred attrs (all)* is end-to-end evaluation on all attributes. The ratio of errors caused by imperfect attribute extraction is nearly double of the previous setup, since sentences containing any of the 59 attributes not covered by our rule-based system all belong to this error class. The ratio of fully correct rules is thereby reduced to 46%. These figures confirm our initial impression that attribute extraction is the main bottleneck of the rule extraction task, but also reveals the sensitivity of the end-to-end task to our ability to detect less frequent attributes and to the quality of value extraction and modality prediction.

	gold attrs		pred attrs (40)		pred attrs (all)	
	#sens	ratio	#sens	ratio	#sens	ratio
wrong attribute set			1352	19.78%	2686	39.30%
correct attr. set, wrong values	1219	17.84%	619	9.06%	409	5.98%
correct attrs, wrong types	152	2.22%	66	0.97%	42	0.61%
only modality is wrong	943	13.80%	1016	14.87%	530	7.76%
fully correct rule	4520	66.14%	3781	55.33%	3167	46.34%

Table 19 Sentence-level evaluation of rule construction from gold attributes and of end-to-end rule extraction

8 Conclusion

This paper documented the process of building the BRISE-Plandok corpus of German building regulations annotated with formal rule representations. A custom rule representation and corresponding annotation guidelines were developed and used for an iterative corpus construction process combining non-expert annotation and expert review, creating a gold standard dataset of more than 7,000 sentences. The annotation process was supported by a rule-based information extraction system using a combination of semantic parsing and pattern matching. This system was also evaluated on the final dataset together with a set of machine learning baselines of varying degrees of interpretability, illustrating the potential of simple and transparent models for information extraction from technical text. All software tools developed for supporting the annotation, review, and evaluation processes documented in the paper are released on GitHub under an MIT license together wih the final BRISE-Plandok dataset.

9 Acknowledgements

Work supported by BRISE-Vienna (UIA04-081), a European Union Urban Innovative Actions project. We thank Judit Ács for advising us on the BERT-based baselines, and the experts of the City of Vienna Building Authority (MA 37 Baupolizei) for their assistance in developing the annotation guidelines. Design of rule representation, development of annotation guidelines: G.R. and B.L. Data preprocessing: G.R. Annotation tools: Á.K. and E.I. Annotation organization: G.R. and E.I. Review of annotations: E.I. Data postprocessing: E.I. Statistics: E.I. Rule-based extraction method and implementation: G.R., Á.K., E.I. Machine learning baselines: E.I. Evaluation: G.R. and E.I. Project coordination: G.R. Writing the paper: G.R. Principal Investigator: A.H.

References

Ahn, K., Bos, J., Kor, D., Nissim, M., Webber, B.L., Curran, J.R. (2005). Question answering with QED at TREC 2005. E.M. Voorhees & L.P. Buckland (Eds.), *Proceedings of the fourteenth text retrieval conference, TREC 2005, gaithersburg, maryland, usa, november 15-18, 2005* (Vol. 500-266). National Institute of Standards and Technology (NIST). Retrieved from http://trec.nist.gov/pubs/trec14/papers/uedinburgh-nissim.qa.pdf

Aires, J.P., Pinheiro, D., Lima, V.S.d., Meneguzzi, F. (2017). Norm conflict identification in contracts. *Artificial Intelligence and Law*, *25*(4), 397–428. Retrieved from https://link.springer.com/article/10.1007/s10506-017-9205-x

Al-Kofahi, K., Tyrrell, A., Vachher, A., Jackson, P. (2001). A machine learning approach to prior case retrieval. *Proceedings of the 8th international conference on artificial intelligence and law* (p. 88–93). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/383535.383545 10.1145/383535.383545

Athan, T., Boley, H., Governatori, G., Palmirani, M., Paschke, A., Wyner, A. (2013). Oasis legalruleml. *Proceedings of the fourteenth international conference on artificial intelligence and law* (pp. 3–12). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/2514601.2514603 10.1145/2514601.2514603

Beach, T., Rezgui, Y., Li, H., Kasim, T. (2015). A rule-based semantic approach for automated regulatory compliance in the construction sector. *Expert Systems with Applications*, *42*(12), 5219-5231. Retrieved from https://www.sciencedirect.com/science/article/pii/S0957417415001360

https://doi.org/10.1016/j.eswa.2015.02.029

Branting, L.K., Yeh, A., Weiss, B., Merkhofer, E., Brown, B. (2018). Inducing predictive models for decision support in administrative adjudication. *Ai approaches to the complexity of legal systems* (pp. 465 – 477). Springer International Publishing.

Curran, J., Clark, S., Bos, J. (2007). Linguistically motivated large-scale NLP with C&C and boxer. *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions* (pp. 33–36). Prague, Czech Republic: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P07-2009

Curtotti, M., & McCreath, E.C. (2011). A corpus of australian contract language: Description, profiling and analysis. *Proceedings of the 13th international conference on artificial intelligence and law* (p. 199–208). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/2018358.2018387

de Marneffe, M.-C., Manning, C.D., Nivre, J., Zeman, D. (2021, 07). Universal Dependencies. *Computational Linguistics*, *47*(2), 255-308. Retrieved from https://doi.org/10.1162/coli_a_00402

10.1162/coli_a_00402

Dimyadi, J., Fernando, S., Davies, K., Amor, R. (2020). Computerising the new zealand building code for automated compliance audit. *6th new zealand built environment research symposium (nzbers 2020)* (pp. 39–46). Retrieved from https://www.buildingbetter.nz/publications/homes_spaces/Dimyadi_et_al_Feb20

Dragoni, M., Villata, S., Rizzi, W., Governatori, G. (2016). Combining NLP approaches for rule extraction from legal documents. *1st workshop on mining and reasoning with legal texts (mirel 2016).*

Fuchs, S., Witbrock, M., Dimyadi, J., Amor, R. (2022). Neural semantic parsing of building regulations for compliance checking. *IOP Conference Series: Earth and Environmental Science*, *1101*(9), 092022. Retrieved from https://dx.doi.org/10.1088/1755-1315/1101/9/092022

10.1088/1755-1315/1101/9/092022

Glaser, I., Moser, S., Matthes, F. (2021). Summarization of German court rulings. *Proceedings of the natural legal language processing workshop 2021* (pp. 180–189). Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.nllp-1.19 10.18653/v1/2021.nllp-1.19

Governatori, G. (2005). Representing business contracts in ruleml. *International Journal of Cooperative Information Systems*, *14*(02n03), 181–216. Retrieved from https://core.ac.uk/download/pdf/14982604.pdf

Governatori, G. (2018). Practical normative reasoning with defeasible deontic logic. C. d'Amato & M. Theobald (Eds.), *Reasoning web 2018* (Vol. 11078, p. 1-25). Springer.

Guo, D., Onstein, E., Rosa, A.D.L. (2021). A semantic approach for automated rule compliance checking in construction industry. *IEEE Access*, *9*, 129648-129660.

10.1109/ACCESS.2021.3108226

Holte, R.C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine learning*, *11*(1), 63–90.

Kalamkar, P., Tiwari, A., Agarwal, A., Karn, S., Gupta, S., Raghavan, V., Modi, A. (2022). *Corpus for automatic structuring of legal documents.* arXiv. Retrieved from https://arxiv.org/abs/2201.13125 10.48550/ARXIV.2201.13125

Kanapala, A., Pal, S., Pamula, R. (2019). Text summarization from legal documents: a survey. *Artificial Intelligence Review*, *51*(3), 371–402.

Karmarkar, N., & Karp, R.M. (1982). *The differencing method of set partitioning.* Computer Science Division (EECS), University of California Berkeley.

Kipper, K., Korhonen, A., Ryant, N., Palmer, M. (2008). A large-scale classification of english verbs. *Language Resources and Evaluation*, *42*(1), 21–40.

Kornai, A. (2010). The algebra of lexical semantics. C. Ebert, G. Jäger, & J. Michaelis (Eds.), *Proceedings of the 11th Mathematics of Language Workshop* (pp. 174–199). Springer. 10.5555/1886644.1886658

Kornai, A., Ács, J., Makrai, M., Nemeskey, D.M., Pajkossy, K., Recski, G. (2015). Competence in lexical semantics. *Proceedings of the fourth joint conference on lexical and computational semantics* (pp. 165–175). Denver, Colorado: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/S15-1019 10.18653/v1/S15-1019

Kovács, A., Gémes, K., Iklódi, E., Recski, G. (2022). POTATO: ExPlainable InfOrmation ExTrAcTion FramewOrk. *Proceedings of the 31st acm international conference on information and knowledge management* (p. 4897–4901). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3511808.3557196 10.1145/3511808.3557196

Kovács, Á., Gémes, K., Kornai, A., Recski, G. (2022). Explainable lexical entailment with semantic graphs. *Natural Language Engineering*, 1–24. Retrieved from https://www.doi.org/10.1017/S1351324922000092

10.1017/S1351324922000092

Kruiper, R., Konstas, I., Gray, A.J., Sadeghineko, F., Watson, R., Kumar, B. (2021). SPaR.txt, a cheap shallow parsing approach for regulatory texts. *Proceedings of the natural legal language processing workshop 2021* (pp. 129–143). Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.nllp-1.14 10.18653/v1/2021.nllp-1.14

Lee, H., Lee, J.-K., Park, S., Kim, I. (2016). Translating building legislation into a computer-executable format for evaluating building permit requirements. *Automation in Construction*, *71*, 49-61. Retrieved from https://www.sciencedirect.com/science/article/pii/S0926580516300796 (The Special Issue of 32nd International Symposium on Automation and Robotics in Construction)

https://doi.org/10.1016/j.autcon.2016.04.008

Malsane, S., Matthews, J., Lockley, S., Love, P.E., Greenwood, D. (2015). Development of an object model for automated compliance checking. *Automation in Construction*, *49*, 51-58. Retrieved from https://www.sciencedirect.com/science/article/pii/S0926580514002155

https://doi.org/10.1016/j.autcon.2014.10.004

Martin, A.D., Quinn, K.M., Ruger, T.W., Kim, P.T. (2004). Competing approaches to predicting supreme court decision making. *Perspectives on Politics*, *2*(4), 761–767. Retrieved 2022-06-30, from http://www.jstor.org/stable/3688543

Miller, G.A. (1995). Wordnet: a lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Modgil, S., & Prakken, H. (2014). The ASPIC+ framework for structured argumentation: A tutorial. *Argument and Computation*, *5*(1), 31-62. Retrieved from http://dx.doi.org/10.1080/19462166.2013.869766

Moens, M.-F., Uyttendaele, C., Dumortier, J. (1999). Abstracting of legal cases: The potential of clustering based on the selection of representative objects. *Journal of the American Society for Information Science*, *50*(2), 151-161. Retrieved from https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-4571%281999%2950%3A2%3C151%3A%3AAID-ASI6%3E3.0.CO%3B2-I

https://doi.org/10.1002/(SICI)1097-4571(1999)50:2<151::AID-ASI6>3.0.CO;2-I

Parent, X., & van der Torre, L. (2013). Input/output logic. D. Gabbay, J. Horty, X. Parent, R. van der Meyden, & L. van der Torre (Eds.), *Handbook of deontic logic and normative systems* (p. 495-544). College Publications.

Recski, G. (2018). Building concept definitions from explanatory dictionaries. *International Journal of Lexicography*, *31*, 274–311. Retrieved from https://academic.oup.com/ijl/article/31/3/274/3835852?guestAccessKey=9f090231-2795-47af-aa44-5fb15d4df0d8

10.1093/ijl/ecx007

Recski, G., Lellmann, B., Kovács, Á., Hanbury, A. (2021). Explainable rule extraction via semantic graphs. *Proceedings of the Fifth Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2021)* (pp. 24–35). São Paulo, Brazil: CEUR Workshop Proceedings. Retrieved from http://ceur-ws.org/Vol-2888/paper3.pdf

Rivest, R.L. (1987). Learning decision lists. *Machine learning*, *2*(3), 229–246.

Saravanan, M., Ravindran, B., Raman, S. (2008). Automatic identification of rhetorical roles using conditional random fields for legal document summarization. *Proceedings of the third international joint conference on natural language processing: Volume-I.* Retrieved from https://aclanthology.org/I08-1063

Shao, Y., Mao, J., Liu, Y., Ma, W., Satoh, K., Zhang, M., Ma, S. (2020). Bert-pli: Modeling paragraph-level interactions for legal case retrieval. C. Bessiere (Ed.), *Proceedings of the twenty-ninth international joint conference on artificial intelligence, IJCAI-20* (pp. 3501–3507). International Joint Conferences on Artificial Intelligence Organization. Retrieved from https://doi.org/10.24963/ijcai.2020/484 10.24963/ijcai.2020/484

Strickson, B., & De La Iglesia, B. (2020). Legal judgement prediction for uk courts. *Proceedings of the 2020 the 3rd international conference on information science and system* (p. 204–209). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3388176.3388183 10.1145/3388176.3388183

Tuggener, D., von Däniken, P., Peetz, T., Cieliebak, M. (2020). LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. *Proceedings of the twelfth language resources and evaluation conference* (pp. 1235–1241). Marseille, France: European Language Resources Association. Retrieved from https://aclanthology.org/2020.lrec-1.155

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., ... Polosukhin, I. (2017). Attention is all you need. I. Guyon et al. (Eds.), *Advances in neural information processing systems 30* (pp. 5998–6008). Long Beach, CA, USA: Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

Waltl, B., Bonczek, G., Scepankova, E., Matthes, F. (2019). Semantic types of legal norms in german laws: classification and analysis using local linear explanations. *Artificial Intelligence and Law*, *27*(1), 43–71. Retrieved from https://link.springer.com/article/10.1007/s10506-018-9228-y

Wrzalik, M., & Krechel, D. (2021). GerDaLIR: A German dataset for legal information retrieval. *Proceedings of the natural legal language processing workshop 2021* (pp. 123–128). Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.nllp-1.13 10.18653/v1/2021.nllp-1.13

Wu, Z., & Palmer, M. (1994). Verb semantics and lexical selection. *32nd annual meeting of the association for computational linguistics* (pp.

133–138). Las Cruces, New Mexico, USA: Association for Computational Linguistics. Retrieved from https://aclanthology.org/P94-1019 10.3115/981732.981751

Wyner, A., & Peters, W. (2011). On rule extraction from regulations. *Legal knowledge and information systems* (pp. 113–122). IOS Press.

Xue, X., & Zhang, J. (2022). Regulatory information transformation ruleset expansion to support automated building code compliance checking. *Automation in Construction*, *138*, 104230. Retrieved from https://www.sciencedirect.com/science/article/pii/S0926580522001030

https://doi.org/10.1016/j.autcon.2022.104230

Zhang, J., & El-Gohary, N.M. (2015). Automated information transformation for automated regulatory compliance checking in construction. *Journal of Computing in Civil Engineering*, *29*(4), B4015001.

10.1061/(ASCE)CP.1943-5487.0000427

Zhang, J., & El-Gohary, N.M. (2017). Integrating semantic nlp and logic reasoning into a unified system for fully-automated code checking. *Automation in Construction*, *73*, 45-57. Retrieved from https://www.sciencedirect.com/science/article/pii/S0926580516301819

https://doi.org/10.1016/j.autcon.2016.08.027