# Improving Legal Information Retrieval: Metadata Extraction and Segmentation of German Court Rulings

Ingo Glaser[a], Sebastian Moser[b] and Florian Matthes
*Chair of Software Engineering for Business Information Systems, Technical University of Munich, Boltzmannstrasse 3, 85748 Garching bei München, Germany*

Keywords: Document Segmentation, Legal Document Analysis, Legal Information Retrieval, Metadata Extraction, Natural Language Processing.

Abstract: Legal research is a vital part of the work of lawyers. The increasing complexity of legal cases has led to a desire for fast and accurate legal information retrieval, leveraging semantic information. However, two main problems occur on that path. First, the share of published judgments is only marginal. Second, it lacks state-of-the-art NLP approaches to extract semantic information. The latter, in turn, can be attributed to the issue of data scarcity. One big issue in the publication process of court rulings is the lack of automatization. Yet, the digitalization of court rulings, specifically transforming the textual representation from the court into a machine-readable format, is mainly done manually. To address this issue, we propose an automated pipeline to segment court rulings and extract metadata. We integrate that pipeline into a prototypical web application and use it for a qualitative evaluation. The results show that the extraction of metadata and the classification of paragraphs into the respective verdict segments perform well and can be utilized within the existing processes at legal publishers.



## 1 INTRODUCTION

The work of legal practitioners is knowledge-intensive and time-consuming. Many studies have shown that legal research is a vital part of the daily work of lawyers (Lastres, 2015; Peoples, 2005). With this, one crucial document type is court rulings. While the legislation defines legal rules, the interpretation of the terms used in law is made through the jurisdiction. That is why legal cases play a crucial role in various legal processes.

As a result, various online databases offering digital access to former cases exist. Many of these databases are hosted by legal publishers. While they aim at providing useful information retrieval features to its users, the actual digitalization process is yet performed in a manual and tedious process.

At first, the legal publishers receive a court ruling via e-mail from the respective court. Now, the court provides the verdict in a simple textual format such as .docx or .pdf. Given any machine-readable format such as Akomo Ntoso (Palmirani and Viatali, 2011), LegalDocML[1], or other private in-house formats, trained employees manually transform the provided verdict into the corresponding target format. This process involves the extraction of metadata and the segmentation of the verdict (see Section 3 for more details). In the next step, a legal author reads through the verdict to gain semantic information about the court ruling. The corresponding information extraction responsibilities range from quite knowledge-intensive tasks such as text summarization to relatively simple tasks such as extracting the area of law.

Despite the existence of online databases, legal information retrieval is not much advanced yet. The reasons for this are manifold. As explained earlier, the digitalization process has to be performed by legal practitioners, which constitutes a bottleneck. Furthermore, valuable semantic information that can be utilized within state-of-the-art information retrieval approaches remains untouched. Instead of manually extracting knowledge about cases, modern Natural Language Understanding (NLU) methods must be applied. This again closes the circle to automating the digitalization as it would provide more extensive datasets that can be used to train machine learn-

[a] https://orcid.org/0000-0002-5280-6431
[b] https://orcid.org/0000-0003-1254-7655

[1] https://www.oasis-open.org/committees/legaldocml



Glaser, I., Moser, S. and Matthes, F.
Improving Legal Information Retrieval: Metadata Extraction and Segmentation of German Court Rulings.
DOI: 10.5220/0010691300003064
In *Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2021) - Volume 1: KDIR*, pages 282-291
ISBN: 978-989-758-533-3; ISSN: 2184-3228

ing (ML) models for specific tasks.

All that results in exhaustive legal research activities for legal workers. Therefore, we want to automate parts of the described process. This paper investigates the feasibility of automatically transforming a court ruling as a court provides it into a machine-readable representation.

## 2 RELATED WORK

In general, legal text is typically conveyed in natural language and not suitable to be processed by computers (Shelar and Moharir, 2018). For that reason, much research concerning knowledge representations of legal documents was performed within the AI & Law community. Particularly representations for legislative and judicative documents were investigated. Akoma Ntoso (Palmirani and Vitali, 2011), defines simple technology-neutral electronic representations in the XML format of parliamentary, legislative, and judiciary documents. Its XML schemas make explicit the structure and semantic components of the digital documents to support the creation of high-value information services. LegalDocML[2] is another standard that is based on Akoma Ntoso. Even the German Federal Ministry of the Interior, with the participation of other institutions, developed a version of LegalDocML tailored to the German legal domain. Ostendorff et al. (Ostendorff et al., 2021) evaluated different document representations for content-based legal literature recommendations.

While great strides have been made in the field of document representations, the use of such representation in an automated digitization process utilizing modern Natural Language Processing (NLP) has remained mostly unexplored. Particularly within the German legal domain, only very little research exists. Structural text segmentation of legal documents was investigated by Aumüller et al. (Aumiller et al., 2021). Based on the assumption that information systems rely on representations of individual sentences or paragraphs, which may lack crucial context, they propose a segmentation system that can predict the topical coherence of sequential text segments. Their system can effectively segment a document and provide a more balanced text representation for downstream applications. Glaser et al. (Glaser et al., 2021) encountered the issue of detecting sentence boundaries in German legal documents. While Sentence Boundary detection (SBD) has been seen as a solved problem for quite some time, domains with solid linguistic characteristics such as the legal domain require tailored models. For that reason, they created an SBD model, trained on German legal documents. In another paper, Glaser and Matthes (Glaser and Matthes, 2020) tried to automate parts of the information extraction part of the publishing process. They compared rule-based approaches and ML approaches to automatically detect the underlying area of law for a given verdict.

Even though there is existing work on legal document segmentation, including metadata extraction (Lu et al., 2011; Lyte and Branting, 2019; Loza Mencía, 2009; Waltl et al., 2019; Chalkidis and Kampas, 2019), they generally rely on existing HTML or XML structure in their input documents. Therefore, they do not generalize to random text inputs without structural features. As a result, to the best of our knowledge, no attempt to transform plain textual verdicts that origin from German courts into a machine-readable format has ever been made before.

## 3 STRUCTURE OF GERMAN LEGAL COURT RULINGS

For this research, we focus on court rulings in civil proceedings as well as criminal law. The court procedure in civil proceedings is regulated mainly by the German civil procedure code (ZPO). As a result, it defines the general structure of a court decision in civil matters. A civil judgment is divided into six parts:

1. Recital of parties (Rubrum): This is the beginning of a court ruling and indicates, in addition to the involved parties and their addresses, the type of the decision, the address of the court, and the case number. While the concrete format of a case number varies from court to court, it always consists of the initials of the court, the processing division of the court, a register number, and a current file number. Sometimes, when being published, the recital of parties contains a verdict title that has been added during the publication process by a legal author. However, the ZPO does not require such a title.
2. Tenor (Tenor): This is the essential part of the judgment, as the dispute is decided here. A tenor usually consists of three different parts, whereas the concrete composition depends on the decision scope. First, the main decision states, for example, whether the defendant must pay the plaintiff the amount claimed or whether the action must or will be dismissed. Second, the possible interest in the claim and the costs of the litigation are con-

[2]https://www.oasis-open.org/committees/legaldocml

sidered. In addition, possibly the question of the provisional enforceability of the judgment (if appeals against the judgment are still possible) must also be decided.

3. Summary of the facts (Tatbestand): The summary of facts contains the central facts on which the case is based. They are presented from the judge's point of view as they were presented in the last hearing. Most importantly, these facts are also the foundation for the final decision.
4. Reasoning (Gründe): Here, the court states its reasoning for the decision made. The reasoning is written in the so-called judgment style, which begins with the result, followed by a gradual justification. If the case at hand is not the first instance, supplementing the court's opinion at hand, the lower court's reasoning is also included. For purposes of distinction, the lower court's reasoning is written in the indirect language.
5. Instruction on the right of appeal (Rechtsmittelbelehrung): Under section 232 of the ZPO, all civil court decisions, unless a representation by a lawyer is required, must contain instructions on how to appeal.
6. Signature of the judges: The final part of the verdict is only a formality and includes the signature of each judge.

A published court ruling usually contains another vital section that the ZPO does not define. That is the guiding principle. In jurisprudence, a guiding principle summarizes the main reasons for a decision by the court. Usually, the judge has written it before a verdict is published. On the other hand, sometimes, this part is referred to as an orientation sentence. Typically, a legal author creates it representing a short text on the court decision, which is more comprehensive than the not always easy-to-understand guiding principle. It offers a classification of the decision and thus provides orientation knowledge, which often cannot be presented by the leading sentences of a decision.

The verdict structure remains for criminal law almost identical. However, the German Criminal Procedure Code (StPO) does not divide the facts and reasoning into two distinct parts but places them into a single reasoning segment. Semantically, of course, the two parts - facts and reasoning - have to be there in this order because a coherent, logical argumentation works this way.

Figure 1 reveals the information we want to extract from court rulings by showing an annotated excerpt of a possible input court ruling for our pipeline. As the figure only includes the first page of a verdict, the remaining pages would include the remaining text segments tenor, facts, and reasoning. Additionally, usually the date and file number of the previous instances are included as well.

# 4 SEGMENTING COURT RULINGS

In the following, we want to discuss how our pipeline converts a textual German court ruling into a structured representation that can then be used for further processing or utilization in an online database. The system is largely written with SpaCy[3], while all neural models are implemented in PyTorch[4]. The initial textual document (e.g., .pdf, .doc, or .docx) is converted to a raw textual representation by use of textextract[5]. In doing so, the system only depends on the textual output of the court ruling, while structural requirements on the input text are low. All the information we infer or create is stored directly in the SpaCy document or with specific tokens and word spans. As a result, it allows us to create a processing pipeline with great flexibility. This adaptability is vital as the pipeline may require changes in the future to suit even more instances.

The performance of text segmentation usually heavily relies on the underlying structures. Therefore, it is important to detect sentences with high accuracy. For that reason, after the space-based tokenization from SpaCy, we use a sentence segmentation system proposed by Glaser et al. (Glaser et al., 2021), specifically tailored to German legal documents. The following segmentation is segregated into three phases: (1) preprocessing of the document, (2) resegmentation, and (3) a final labeling step. The possible text segments that can be assigned are GUIDING_PRINCIPLE, TENOR, FACTS, and REASONING. Furthermore, a verdict also has many meta segments, namely PRE_TITLE, TITLE, COURT, DATE, SOURCE, KEYWORDS, DECISION_TYPE, PREVIOUS_INSTANCES, NORM_CHAIN as well as IGNORE and UNKNOWN, which are used for any case not fitting in this taxonomy. Initially, each sentence has the segment label UNKNOWN.

## 4.1 Preprocessing

Retokenization is the first significant step necessary due to the different formatting styles in a court ruling. Different fonts, large spaces between letters of one

[3] spacy.io
[4] pytorch.org
[5] github.com/deanmalmgren/textract

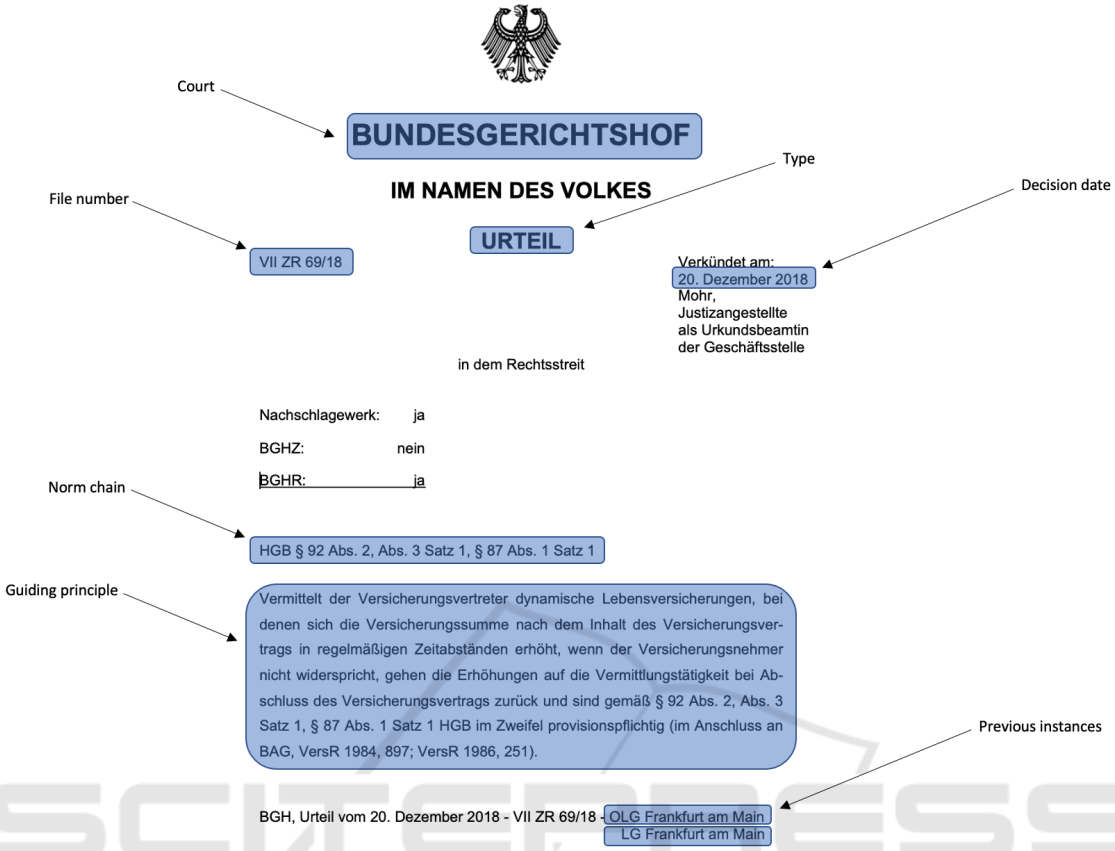


Figure 1: Excerpt of a verdict from the German supreme court (BGH) with annotated metadata and segmentation.

word, or other formatting choices are not uncommon and introduce some minor problems tackled in this step. Most problems occur with the beginning formulation of the tenor. Terms like ”für Recht erkannt” (found for right) or ”beschlossen” (decided) are written with large spaces between the letters. During this step, we remove spaces between the letters and assign the TENOR segment for the following sentence if we find such a formulation.

The next preprocessing step constitutes the extraction of all structural components, such as headlines and enumerations. Our system can match 5 different enumeration types: (1) roman (*I.*, *II*, ...), (2) alphabetic (*a)*, *B*, ...), (3) numeric (*1.*, ...), (4) combined (*1a*, ...), or (5) marginal numbers/RVs (*1*). Each type is matched via its own set of regular expressions. Any matching token is then checked for validity. Those validity checks include, among others, whether it occurs at the start of a line or whether the token before is an enumeration. Given the sequence of all such tokens, we now check that all enumeration sequences are well-formed, i.e., they have the correct start token, each token in a sequence has the same punctuation, and they are correctly nested. The algorithm is independent of any writing style of enumerations, allowing variations such as ”1.”, ”1)”, or ”1.)”. Only the global validation takes the writing style into account when linking the different enumerations.

After extracting all structural components, we consider the different headlines. German court rulings only contain a handful of different headlines, which are all solid indicators for the segment of the following sentence. The only exception here is the headline ”Gründe” (Reasoning), as it can be commonly found before FACTS or REASONING segments (see Section 3). If there are no more specific headlines (”Tatbestand” (Facts), or ”Entscheidungsgründe” (Decision reasoning)), the sentence after ”Gründe” (Reasoning) will be REASONING for the moment. To ensure that each matched word is indeed a headline, we check the characters before and after the found token (for instance, at the beginning of a line). Furthermore, we filter the headlines, such that at most, one for each segment type is found. Last but not least, page numbers and other unnecessary spans of text are removed.

As each court has its formatting and writing style, we introduce a court-specific pipeline component in

the following step. First, every relevant piece of information for that specific court is matched. Second, we use this information in a post-processing step to reason about the segment of specific sentences. To test this, we implemented a pipeline component for BGH court rulings, as they have a unique structure with their mixture of norm chains and guiding principle sentences at the beginning. Due to the flexibility of our pipeline, it is straightforward to extend it for other courts, as this might be necessary for court rulings with esoteric formatting.

### 4.2 Resegmentation

Based on the headlines and enumeration information, we now assign new segment ends and starts, such that the headlines and enumeration symbols are treated as their own sentences. This step is necessary to separate structural information from content.

### 4.3 Labeling

We will now assign the final segment labels to each sentence in the document based on the extracted information. Additionally, we classify each sentence individually via a BERT-based classifier (Devlin et al., 2019). The predicted label is used in the following as a second measurement to determine the actual segment label of each sentence. If we have not yet assigned a label by any other rule, we will use the classification output.

We use the bert-base-german-cased pre-trained BERT model as the base for our classifier, as it was trained partially on German legal documents. On top of BERT, we put a linear classification layer taking the pooled output of BERT. The classifier is trained with Flair (Akbik et al., 2019). As a dataset, we used 73k German court rulings in an XML format which contains segment information for each sentence. In a standalone evaluation, the classifier (97.65 Macro $F_1$ on test set; only trained on text segments) showed the unwanted behavior of switching back and forth between segment labels within a continuous span of sentences. This, together with the fact that we cannot train a classifier for every segment due to a lack of available data, was why that we use the classifier as one pipeline component that produces the segment annotations and the rest.

The next step is to consistently label the sentences between two found headlines known to be commonly found in that order, i.e., if the following headline after a facts headline is for the reasoning segment, we know for sure that all sentences in between are FACTS. If we have a title headline, everything before will be annotated PRE_TITLE. That is a simplification because courts sometimes add information before the title, while those segments have always a headline and thus are already annotated. Afterward, we will smooth the segment annotations such that no two-segment classes are found interleaved in the document.

Eventually, one final segmentation step is only applied if we have not found any FACTS. This is often the case due to the ambiguous "Gründe" (Reasoning) headline. In such cases, we use a heuristic based on the enumerations. If the reasoning block starts with a roman *I* or alphabetic *A* enumeration, we will annotate everything up to the following enumeration token (*II* or *B*) as FACTS. Some verdicts are outliers, but the practice has shown that this rule is correct in almost all cases.

## 5 EXTRACTING METADATA

After segmenting the court ruling into its distinct components, we take care of extracting metadata. At first glance, it may seem not very reassuring to extract metadata after the segmentation. However, the segments play a crucial role for the extraction of metadata as it defines where the required information can be found.

In the information extraction step, we identify all reference numbers, the specific file number for a verdict, all referenced courts, the concrete court of the verdict, dates and the specific date of the decision, category of the court ruling, norm chains, and previous instances. This procedure is again a two-step process as some pieces of information are needed for other steps. We need, for example, all reference numbers before being able to identify previous instances. Thus each preprocessing step (first paragraph of each metadata part) is finalized before any of the postprocessing steps (second paragraph) can be performed.

### 5.1 Reference Number

The file numbers for civil cases and criminal cases vary in their syntactic. For that reason, based on two regular expressions, we identify the reference numbers of the following forms: (1 - ZPO) *Prefix Department/Chamber/Senate Register-Reference Year.Number Suffix* and (2 - StPO) *Department/Chamber/Senate RegisterReference Number/Year*. However, some specific courts may add additional elements to the beginning or trailing of the base form.

Afterward, we parse the matched spans of text and add potential additional prefixes or suffixes. Next, we

extract the reference number with the highest instance level as the reference number of the current court ruling. With this, we only search in non-text segments as those sometimes contain references to court rulings of higher courts. In the rare case that no references are found, the list of excluded segments is reduced until one reference number is found. Each *RegisterReference* has a specific legal meaning, and it is possible to assign an instance level to each of them, i.e., for every *RegisterReference* we collect the possible instance-level this *RegisterReference* is used in. If multiple levels are possible, we used the lowest one. Based on the *RegisterReference* we also heuristically extract the code of procedure of the verdict (ZPO or StPO).

### 5.2 Courts

In order to detect the court of a verdict, we utilize a dictionary lookup. Therefore, a dictionary of all German courts was created by crawling respective online resources. The dictionary may contain the same court multiple times, as we use the different possible abbreviations as a key. Based on that dictionary, we annotate each court found in the document and assign an instance level to each of them.

In the next step, similarly to the rules for the file number above, we choose the court with the highest instance level to be the court of the given verdict. With that, again, we only consider courts that are outside of a text segment and only reduce the list of excluded segments if no court was found.

### 5.3 Dates

The extraction of the promulgation date can be considered a relatively simple task. After matching all dates in the court ruling via regular expressions, the latest date is chosen. However, only dates outside of the guiding principle, the facts, and the reasoning of the verdict are considered. Eventually, the date is converted to the ISO format.

### 5.4 Type of Verdict

The different types of court rulings in Germany are pretty limited. For that reason, we utilize a predefined list of words (”Beschluss”, ”Urteil”, ”Teilurteil”, ”Leitsatzentscheidung”, etc.). Each token of sentences segmented into the recital of parties (Rubrum) is matched against that list. As the verdict type is always defined at the beginning of the decision, the first match is chosen as the respective category.

### 5.5 Previous Instances

The extraction of previous instances and their file numbers and dates is done only through postprocessing steps. After obtaining all reference numbers, except the one already classified as the file number for the given verdict, we look for the most extended sequence of reference numbers that each has at most one line between them. This assessment is made, as the previous instances always occur together. Then those are parsed and potentially identified as previous instances. Finally, we identify where the court and date for each of them are (i.e., before, after the reference number) and extract this information.

### 5.6 Normchain

To extract the norm chain, firstly, tokens that are commonly found in norms are matched. To do so, we extracted all norms from the norm chains in our classification dataset. The resulting norms were split into their components, such as words, register characters, numbers, and punctuation. All lines containing a match that contains at least 95% of such tokens are stored as a potential norm chain. Then, the most extended continuous sequence of norms is identified and classified as the norm chain of the verdict. Furthermore, the court-specific pipeline component provides specific information as well, as some courts do not have continuous norm chains.

## 6 EVALUATION

The proposed system covers various tasks that differ in their characteristics. Some tasks, such as extracting the case number, could be quickly evaluated quantitatively with standard metrics such as precision, recall, and $F_1$. On the other side, text segmentation requires different evaluation methods as a segment can be partially correct. Furthermore, tasks such as the extraction of norm chains require even some qualitative feedback from domain experts. In order to be able to assess the overall performance of our system, we came up with a custom evaluation method combining qualitative and quantitative measures.

Before the remainder of this section elaborates on the essential criteria, on the evaluation itself, and on error analysis, we introduce Verlyze. Verlyze is a web application implementing our proposed pipeline. Users can upload original verdicts in various input formats as provided by courts. During the upload, the court ruling is processed by that pipeline. Verlyze performs further semantic analysis, which is not part

of this paper. The structured, machine-readable representation is then stored in a database, enabling legal information retrieval. Figure 2 shows a screenshot of the verdict view after retrieving a specific verdict. A user can scroll through the different text segments, highlight references, read through meta information, or even inspect semantic information. Furthermore, the original document can be shown in order to allow a quicker assessment during the evaluation.

## 6.1 Description Criteria/Grading System

We used 50 randomly selected German court rulings chosen from a larger dataset of approximately 800 verdicts for the evaluation. A legal publisher created the dataset. However, they provided it to us only after finalizing the system in order to avoid overfitting. Thus, none of those court rulings were used for testing the implementation. To ensure a variety of different documents, the random selection was further subdivided by instance level. Twenty court rulings are from the BGH, fifteen from local supreme courts (OLG), and fifteen from other courts.

The evaluation was done by four evaluators, two of the authors, and two employees from the legal publisher that are skilled in the publication process of court rulings. The evaluation task was to evaluate the metadata extraction and the segmentation for each selected verdict. The annotators could give up to 10 points in each category, with 10 points denoting a perfect result. The categories include *Guiding Principle*, *Tenor*, *Facts*, *Reasoning*. Each of those text categories should be evaluated individually based on the content. For example, if the system wrongly assigns all the facts to the reasoning part, while the reasoning part is otherwise perfectly extracted, the facts-score should be 0, and the reasoning-score should be 10. For such cases, we also added a *Structure* category, which allowed annotators to judge the formatting, extraction of enumerations, and overall structure.

For the norm chain, the annotators needed to judge if all *Norm*s were extracted as well as the more fine-grained extraction of the *Paragraphs*. Similarly for the previous instances, the evaluation included the *Courts*, their *Reference Numbers* and their *Dates*.

For the basic meta information (*Id* or reference number, *Ruling Court*, *Ruling Date*) we choose a binary score as the result has no variability in the correctness. We subdivided the criteria court into one point for the instance level and one for the correct place.

If any of this information was not present in the original document, the annotators had to denote an $x$ instead. The score for each category shall be based on how much wanted content is present in the processed representation. The annotators also provide a *Total* score for each verdict. To analyze the scores, they were normalized to represent a percentage.

## 6.2 Annotator Agreement

As the scoring for a category can be subjective, and there are sometimes no hard correctness rules, we will look at the inter-annotator agreement. Usually, the inter-annotator agreement is used to assess the quality of labels in a dataset. However, we argue it can be used for our purposes as well. In the following, the annotators will be called A0 to A3. When looking at the mean scores per category, as seen in Figure 3, annotators A0, A2, A3 have very similar annotations. The differences between their scores for each category are negligible and only differ in single percentages (except for *Reasoning*, but this shall be discussed in Section 6.4). A1 has assigned more pessimistic scores but surprisingly gives a higher total score compared to A0. The range for the mean *Total* per annotator is between 69.4% and 83.2%. For each category and verdict combination, we extracted the scores to calculate the Fleiss' kappa (Fleiss, 1971). We then use that score to assess the inter-annotator agreement. The Fleiss' kappa for the annotators is 55.6%, which denotes a moderate to substantial agreement. This measure is not perfect in our case. However, the Fleiss' kappa does not take into account the difference between annotation scores. As a result, the combination of the Fleiss' kappa score together with the small absolute differences between annotators suggests a proper evaluation.

## 6.3 Evaluation

Moving to the general evaluation, we see overall high scores in Table 1. The worst results are perceived for the extraction of the previous instances with a high standard deviation. This indicates that the extraction either works or does not work, but there is no middle ground. The subsequent highest variability can be found for *Id*, *Court*, and *Date*. This fact has to relate to the fact that they are a "binary variable". Thus a high variability is expected.

In contrast, identifying the text segments works very well. They have the highest mean scores and the lowest variability. This was also expected as the meta categories have more potential for variability solemnly based on how their information is presented, e.g., there are numerous ways to reference one specific norm in a norm chain.

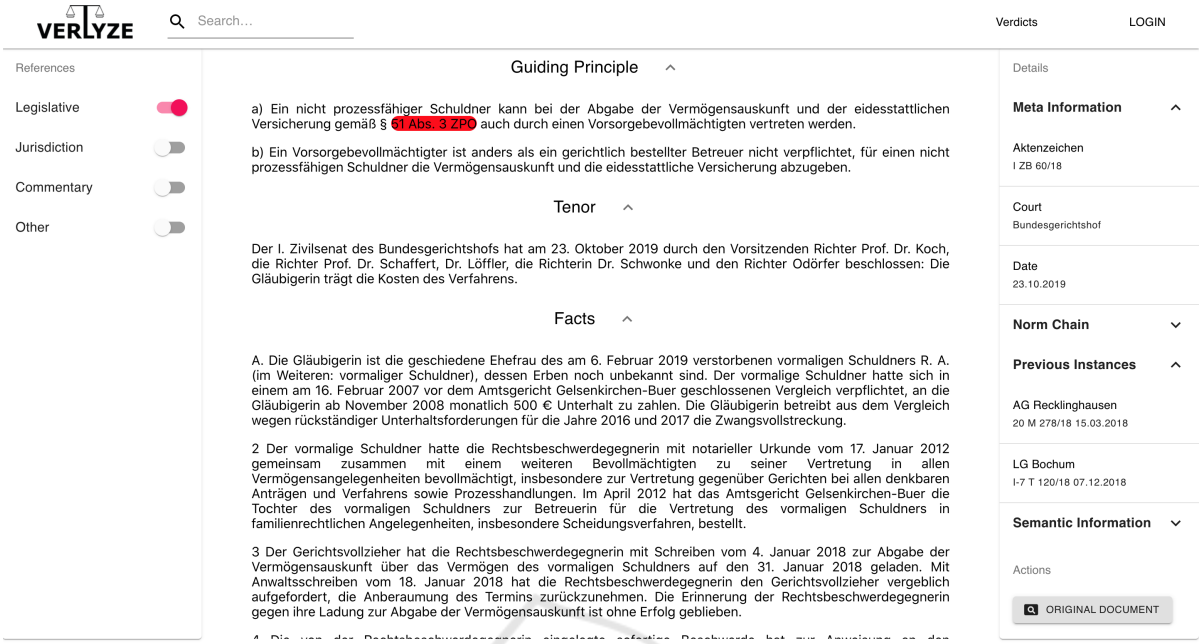


Figure 2: Screenshot of our web application implementing the proposed pipeline.

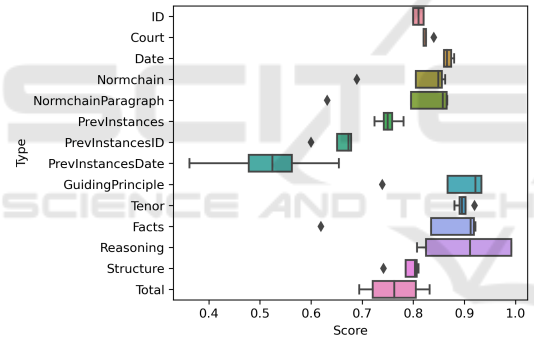


Figure 3: Range of the mean scores per category for each annotator.

To quantify the skew in the dataset, we also calculated the mean score per verdict and then reported the median of those means in Table 1. In all cases, the median is higher than the mean score, in some even substantially. With this in mind, it is evident that our system produces very good to perfect results for all categories in most cases, but some outlier verdicts heavily influence our scoring as they give a poor result. In the following, we want to specifically look at those outliers, i.e., the verdicts with the lowest mean scores in each category.

## 6.4 Error Analysis

The error analysis for outlier verdicts is straightforward in our system, as most errors are introduced by a failure of a specific component in the processing pipeline. We will now move through each category, determining where our pipeline introduced an error and how they can be fixed.

Table 1: Mean, median score and standard deviation for each category. Std can be interpreted as the variability of extraction results for the different verdicts.

| Category | Mean | Median | Std |
|---|---|---|---|
| ID | 81.0% | 100% | 39.3% |
| Court | 82.5% | 100% | 38.1% |
| Date | 86.8% | 100% | 33.9% |
| Normchain | 81.9% | 100% | 34.2% |
| NormchainParag | 81.2% | 95.0% | 33.9% |
| PrevInst | 75.3% | 100% | 38.8% |
| PrevInstID | 65.9% | 96.3% | 43.1% |
| PrevInstDate | 49.5% | 50.0% | 44.5% |
| GuidingPrinciple | 88.8% | 100% | 30.4% |
| Tenor | 89.8% | 100% | 27.2% |
| Facts | 84.7% | 96.3% | 31.6% |
| Reasoning | 91.1% | 92.9% | 21.3% |
| Structure | 79.2% | 81.3% | 13.5% |
| Total | 76.4% | 80.0% | 19.9% |

For the text segments *Facts* and *Reasoning*, the problem lies in the segmentation algorithm as in some cases, a differentiation is complex without an excellent semantic understanding of the German language. To solve this, we would need to use a more suited language model for our domain, and the reliability of its classification needs to be increased. For the *Tenor* and *Guiding principle*, we have a similar problem (differ-

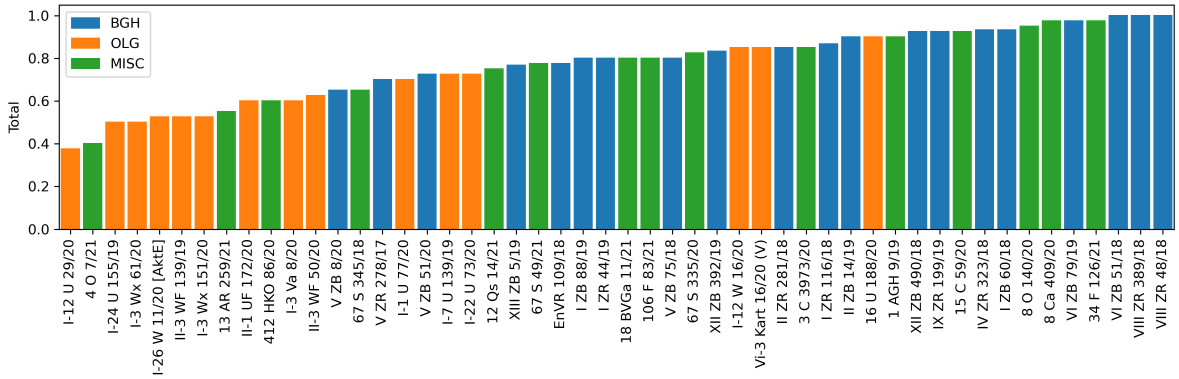


Figure 4: Total mean scores for each verdict with color annotation on the given court type.

entiating them from the other text segments), but their errors might be solved by introducing a sophisticated parsing of the recital of parties. The rubrum has very distinct pieces of information, and their classification will further help determine the type of a specific text segment. This way could better identify the text segments that come before or within the rubrum. Also, the rubrum is sometimes segmented together with the facts or reasoning segment, which introduces more problems to further steps in the pipeline. Most of the low scores for the *Structure* category can also be attributed to this.

For the *Previous instances* and *Id*, the major problem is the differentiation between the reference numbers. Using the instance level induced by the reference number to order them is an insufficient heuristic as there are some edge cases for which this does not hold. Taking the Date or their text position into account might be necessary (as both are commonly found at the beginning of the text). There are also some cases for *Previous instances* where the court was previously unknown (e.g., different writing), which can be solved by extending the court dictionary.

*Court* and *Date* have similar problems as either a higher precedence court or an earlier date is found in a non-text segment. In one case, the segmentation was the reason for a faulty extraction, as the segmentation algorithm combined the rubrum with the reasoning segment, and thus a court from a different segment with higher search precedence was used. Here it is necessary to take the context a piece of information is found more into consideration. Both are not found within a paragraph, and there are standard formulations within their context.

There are three types of edge cases for the *Norm chain*: (1) the norms are within the text and are not further formatted in a specific way, (2) they contain more extended expressions of unknown texts, and (3) they contain different words afterward which are uncommon. To solve the first edge case, we would need to extend the extraction of norms to the whole text and then classify them, which can be done relatively straightforward with modern ML tools. For the latter two cases, we would need to extend our dictionary with uncommon norm terms. However, we have to say that there always will be a missing term due to the nature of the German language.

The overall scores for *Total* are depicted in Figure 4. In the worst cases, the segmentation is insufficient, and consequently, other errors accumulate. This fact further shows the necessity to introduce a more reliable and semantically informed segmentation. We also investigated the *Total* mean scores per court type in the dataset. BGH court rulings work best with 85.9%, but this was expected as we have specifically created a pipeline component for them. Surprisingly, OLG court rulings have the worst total score with 63.5%, compared to 78.2% for other courts. Identifying the guiding principle and the previous instances is hard for OLG rulings, which might be the reason for their low scores. This might have to do with the fact that many of the tested OLG cases have additional information at the beginning which is not following a consistent structure.

## 7 CONCLUSION & OUTLOOK

This work examined the possibility of automating the court ruling publishing process for the German legal domain. A state-of-the-art language model, namely BERT, with a classification head on top, was fine-tuned to classify sentences into the corresponding verdict components. Furthermore, different verdicts from various courts were examined to implement rule-based approaches and heuristics combined with the trained model to automatically provide a pipeline capable of transforming court rulings from various in-

put sources. We could show that it is feasible to extract metadata and segment court rulings with great accuracy.

Nonetheless, this research contains some limitations. While we utilized court rulings from different sources and instances, the system, particularly the court-specific rule-based modules, was tuned based on our inputs. Even though we evaluated the proposed approach on unseen court rulings, even from small courts, verdicts from courts of different jurisdictions (financial, social, employment) may worsen the results as their structure might be different. However, the whole pipeline is implemented in an extensible manner so that it is easy to enhance the rules to match other inputs.

Another promising approach may be the incorporation of a different head on top of BERT. Specifically, instead of classifying the whole sequence based on the pooled representation, adding a linear layer on top of the hidden-states output might be interesting to compute span start logits and span end logits. The model would only be responsible for defining the start and end of each segment instead of classifying each sentence. Due to the nature of such a token-based classification task, it may be feasible for our classification task.

While most of our rule-based and heuristic approaches seem to be adequate, it is worth investigating in the future whether modern language models can help to classify tokens with respect to some of the metadata that did not perform well for us, such as the previous instances of the court ruling. This could even improve our reported results further.

Last but not least, we implemented our pipeline in a prototypical web application called Verlyze, allowing the research community to build even more reliable systems on top of our implementation.

## REFERENCES


Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., and Vollgraf, R. (2019). Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Aumiller, D., Almasian, S., Lackner, S., and Gertz, M. (2021). Structural text segmentation of legal documents.

Chalkidis, I. and Kampas, D. (2019). Deep learning in law: early adaptation and legal word embeddings trained on large corpora. *Artificial Intelligence and Law*, 27(2):171–198.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. In *Psychological Bulletin*, volume 76(5), pages 378–382.

Glaser, I. and Matthes, F. (2020). Classification of german court rulings: Detecting the area of law. In *ASAIL@ JURIX*.

Glaser, I., Moser, S., and Matthes, F. (2021). Sentence boundary detection in german legal documents. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART*, pages 812–821. INSTICC, SciTePress.

Lastres, S. A. (2015). Rebooting legal research in a digital age.

Loza Mencía, E. (2009). Segmentation of legal documents. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 88–97.

Lu, Q., Conrad, J. G., Al-Kofahi, K., and Keenan, W. (2011). Legal document clustering with built-in topic segmentation. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 383–392.

Lyte, A. and Branting, K. (2019). Document segmentation labeling techniques for court filings. In *ASAIL@ ICAIL*.

Ostendorff, M., Ash, E., Ruas, T., Gipp, B., Moreno-Schneider, J., and Rehm, G. (2021). Evaluating document representations for content-based legal literature recommendations. *arXiv preprint arXiv:2104.13841*.

Palmirani, M. and Vitali, F. (2011). *Akoma-Ntoso for legal documents*, pages 75–100. Springer.

Peoples, L. F. (2005). The death of the digest and the pitfalls of electronic research: what is the modern legal researcher to do. *Law Libr. J.*, 97:661.

Shelar, A. and Moharir, M. (2018). A comparative study to determine a suitable legal knowledge representation format. In *2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, pages 514–519. IEEE.

Waltl, B., Bonczek, G., Scepankova, E., and 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.