Risks and Limits of Automatic Consolidation of Statutes

Max Prior and Adrian Hof and Niklas Wais and Matthias Grabmair
Technical University of Munich
Boltzmannstraße 3
85748 Garching near Munich, Germany

Abstract

As in many countries of the Civil Law tradition, consolidated versions of statutes – statutes with added amendments – are difficult to obtain reliably and promptly in Germany. This gap has prompted interest in using large language models (LLMs) to 'synthesize' current and historical versions from amendments. Our paper experiments with an LLM-based consolidation framework and a dataset of 908 amendment–law pairs drawn from 140 Federal Law Gazette documents across four major codes. While automated metrics show high textual similarity (93–99%) for single-step and multi-step amendment chains, only 50.3% of exact matches (single-step) and 20.51% (multi-step) could be achieved; our expert assessment reveals that non-trivial errors persist and that even small divergences can carry legal significance. We therefore argue that any public or private deployment must treat outputs as drafts subject to rigorous human verification.

1 Introduction

Legal systems of the Civil Law tradition are based on statutes. Statutes change over time. Changes are typically ordered by the legislator and implemented via other (amending) statutes; these amendments, published in the official gazette, describe how the current wording of a statute is to be changed, but do not spell out its updated 'version'. This makes the process of consolidation necessary, where amendments are used to update the text of a statute in order to get the current 'version'.

Access to consolidated versions of German statutes is limited. Non-legally-authoritative platforms provide current texts and separate, authoritative amendment logs, but consolidated texts may appear with substantial delays. Users must often reconcile amendment logs with outdated consolidations – a process that is time-consuming and error-prone for specialists and non-specialists alike. For illustration, the Act on Data Protection and the Protection of Privacy in Telecommunications and Telemedia, effective 14 May 2024, was not integrated into the consolidated text by 30 July 2025.[1] When courts apply outdated statutory provisions, the consequences can be significant: In October 2019, the Higher Administrative Court of Baden-Württemberg (Germany) prohibited evening and Sunday afternoon matches in SC Freiburg's new football stadium, relying on noise limits that had already been superseded by a revised regulation since September 2017. The ruling was later challenged because the applicable building permit of November 2018 should have been assessed under the updated regulation, which allowed five extra decibels.[2]

A second, more fundamental access problem is the lack of historical consolidated versions. Reliable versioning is essential: In criminal law, courts must compare the law at the time of the offense with the law at the time of sentencing and apply the more lenient provision (*lex mitior*). Without reliable access to historical texts, courts and counsel face unnecessary uncertainty, potentially affecting the rights and liberties of the accused. When solely relying on authoritative sources, statutes need to be rolled back based on prior amendments that have been published in the official gazette.

Our Contribution. The described gaps and practical needs create pressure to automate the process of consolidation. Automated consolidation research spans rule-based pipelines to machine learning and recent generative approaches. Prior systems demonstrate that computable amendment operations are feasible but also reveal the fragility of templates and the sensitivity to document quality. Given the task

[1] https://www.gesetze-im-internet.de/ttdsg/TTDSG.pdf

[2] https://www.lto.de/recht/hintergruende/h/vgh-bawue-3s147019-sc-freiburg-stadion-laerm-immissionsschutz-anwohner-bundesliga

Proceedings of the Natural Legal Language Processing Workshop 2025, pages 396–407
November 8, 2025

of applying commands written in natural language to a text, this literature motivates experimentation with LLMs. We present the first LLM-based approach to consolidating German law, addressing a critical gap in legal infrastructure where historical versions are unavailable and current consolidations face substantial delays, and an in-depth analysis of its benefits and shortcomings. Our contributions are:

1. **Dataset**: We compiled 908 amendment-law pairs from 140 Federal Law Gazette (German: Bundesgesetzblatt) PDFs, aligning them with consolidated laws from 2019-2025. This benchmark dataset captures complex legal changes and can be continuously updated with new amendments to test how well historical law can be reconstructed.

2. **Framework**: Our automated consolidation framework utilizes GPT-4.1-mini to apply amendments to existing laws. The system handles both single amendments and, as a novelty, multi-step chains (averaging 2.79 amendments per chain).

3. **Evaluation**: We evaluate our system to investigate how well LLMs handle the task of consolidation. Our setup reveals low and highly variant exact match rates ranging between 2.36% and 75.93%, and a semantic similarity of 93-99% for four core legal codes (Civil, Criminal, Commercial, and Income Tax). Expert review of 100 imperfect consolidations revealed that 51% of errors had minimal to moderate impact, with 78% requiring only trivial corrections. We also encounter difficulty in reliably identifying ground truth versions of certain codes at different time points.

4. **Prototype**: We developed a web application that demonstrates practical deployment, enabling users to access and view historical law versions since 2019 (extendable to 1949) through an interface that processes amendments and creates version.

The remainder of this paper is organized as follows. Section 2 reviews Germany's current legal infrastructure and automated consolidation research from rule-based systems to machine learning approaches. Section 3 describes the creation of our dataset, which involves extracting amendments from Federal Law Gazette PDFs and aligning them with consolidated law versions from 2019 to 2025. Section 4 presents our experimental setup for single-steps and multi-step amendment consolidation. Section 5 evaluates the framework through automated metrics and expert legal assessment. Section 6 shows our user interface prototype intended for public experimentation. Finally, Section 7 summarizes our contributions as the first LLM-based approach to German law consolidation and discusses future directions.

2 Background

German legal professionals lack an authoritative archive of historically consolidated federal statutes. "Laws on the Internet" ("Gesetze im Internet")[3] provides current federal laws without historical versions. Amendments have been published since 1949 in Federal Law Gazettes (German: Bundesgesetzblatt)[4], and since 2023 on recht.bund.de,[5], but these publish only the amending texts, not integrated consolidations. Private efforts such as buzer.de[6] partially fill the gap (post-2006 snapshots), yet coverage and timeliness remain limited. Commercial platforms offer code version comparisons, but having a openly available consolidated version history remains desirable.

Automated legal consolidation has evolved from rigid rule-based systems to flexible machine learning methods. Arnold-Moore (1997); Arnold-Moore (1995) pioneered this field with a specialized drafting environment where editors modified statutes while if-then heuristics captured edits as machine-readable logs, enabling automatic consolidation. This established that amendments could be computationally processed rather than manually applied.

Ogawa et al. (2008) advanced this by eliminating specialized environments. They parsed amendments directly from published Japanese Acts, extracting structured operations from natural language descriptions and converting them into formal operations. This enabled the processing of even pre-digital amendments, allowing for the complete reconstruction of the timeline. Using just sixteen regular expressions, their system achieved 99.47% accuracy—proving automated consolidation could match human precision.

[3] https://www.gesetze-im-internet.de/
[4] https://www.bgbl.de/
[5] https://www.recht.bund.de/
[6] https://www.buzer.de/

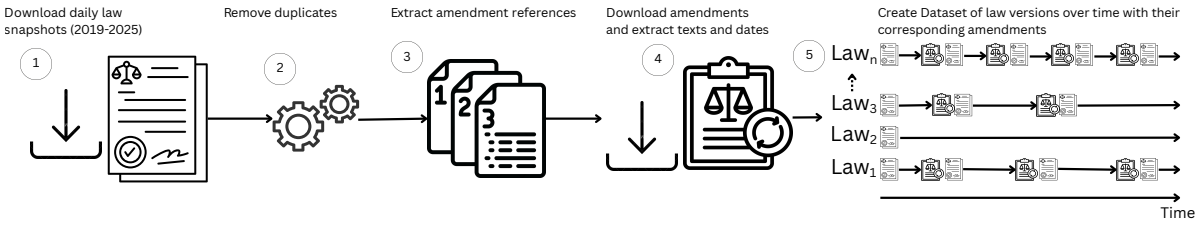

Figure 1: The multi-step process of dataset creation, from left (1) to right (5).

Adapting this approach to Greek legislation exposed significant challenges. Garofalakis et al. (2016) developed a comprehensive pipeline that spans from downloading amendment PDFs to publishing consolidations online. They enhanced the rule-based approach with statistical preprocessing to identify characteristic verbs (add, delete, substitute) since Greek legal language varies more than Japanese. Despite these improvements, they initially achieved only 37.1% accuracy, but this increased to 59.4% with manual corrections. This performance gap stemmed from data quality differences—the Japanese system processed structured XML while the Greek system handled inconsistent PDFs.

These limitations motivated Fabrizi et al. (2021) to adopt machine learning, reframing amendment classification as token labeling, where models learn which words signal different types of change. Rather than manually coding patterns for every variation, their system learned from examples. This eliminated rigid rules and improved robustness to language variation. Unlike Ogawa et al. (2008) and Garofalakis et al. (2016), who needed manual intervention for unexpected templates, this approach was adapted through retraining.

Beyond specialized drafting environments, standardization efforts provided crucial infrastructure. Palmirani and Vitali (2012) developed Legislative XML principles, establishing machine-readable formats amendment processing and temporal versioning. This structured approach proved essential—systems using well-formed XML consistently outperform those processing unstructured PDFs, as the contrasting results between Japanese and Greek implementations would demonstrate.

Etcheverry et al. (2024) introduced the first generative model for legal consolidation in French law, treating it as a text generation task. Given an initial law and amendment, their model generates the complete consolidated text. They created datasets of triplets (initial versions, amendments, ground truth) for systematic evaluation, shifting from rule-based classification to end-to-end generation. However, two critical limitations restrict practical deployment. First, context window restrictions prevented processing over half of the real amendments—complex amendments with substantial changes exceeded input limits. Second, the system handled only single amendments, not the sequential chains typical in real legislation, where amendments build on previous changes over decades. These constraints reveal the gap between current capabilities and requirements for reconstructing complete legislative histories.

3 Data

Figure 1 shows our pipeline for creating the dataset to track the 'evolution' of selected German federal statutes over time.

Step 1 downloads daily snapshots of all federal laws from the Laws on the Internet repository[7] covering 2019 to 2025. Each snapshot contains approximately 300 MB of XML files representing all federal legislation. Since most laws remain unchanged on a daily basis, this raw data contains significant redundancy.

Steps 2 and 3 identify and preserve only meaningful changes. We process each law's XML files in chronological order, comparing consecutive versions to detect any modifications. When identical content appears across multiple days, we keep only one version. This de-duplication retains all substantive amendments while reducing storage requirements. Each preserved version corresponds to a specific amendment that altered the law's content.

Step 4 links these law versions to their official sources. The Federal Law Gazette serves as Germany's official publication for amendments, providing authoritative texts of amendments. We match each detected change to its corresponding Gazette entry. We use GPT-4.1 with a structured prompt

[7] https://github.com/QuantLaw/gesetze-im-internet

(see Appendix A) to extract amendment text and effective dates.

Step 5 assembles all components into our complete dataset. We merge the consolidated law versions with their corresponding amendments and effective dates to create a temporal record. As Figure 1 shows, this process reveals distinct patterns: Law_1 underwent three amendments during our study period, Law_2 remained unchanged, while Law_3 and Law_n experienced varying numbers of modifications at different times. This dataset enables precise tracking of how each law evolved throughout the examination period.

3.1 Legal Code Selection

While our framework can be applied to any German law, resource constraints motivated a focused evaluation of four foundational codes spanning civil, criminal, commercial, and tax law. This selection concentrates on high-impact, frequently consulted domains with diverse amendment patterns. The Civil Code (German: Bürgerliches Gesetzbuch, BGB) governs private relations, including contracts, property, family, and inheritance, and contains 177 paragraph-level comparisons. The Criminal Code (German: Strafgesetzbuch, StGB) comprises 162 provisions that define offenses and penalties. The Commercial Code (German: Handelsgesetzbuch, HGB) regulates business transactions and corporate law with 254 provisions. The Income Tax Act (German: Einkommensteuergesetz, EStG), the central tax statute, is highly complex and frequently amended, contributing 315 comparisons. These codes yield a dataset of 908 comparisons.

4 Experiments

Using the dataset, we tested whether we can reconstruct law versions by applying amendments to initial versions. We evaluate automated legal text consolidation in single-step (isolated amendments) and multi-step (sequential modifications over time) setups.

In single-step experiments, we apply one amendment to an initial version to create a predicted version, then compare it against the actual version using similarity scores. In multi-step experiments, we apply n amendments sequentially to an initial version and compare the final predicted version against the exact version after n changes.

This multi-step approach serves two purposes: it reduces computational costs by requiring fewer similarity calculations and ,more importantly, it validates whether laws can be reconstructed accurately when intermediate versions are unavailable—a common scenario in practice. While single-step is straightforward—applying one amendment to produce a predictable result—multi-step processing involves challenging dependency chains. If Amendment 1 is not used correctly, Amendment 2 cannot, e.g., locate the text "10,000 euros" because this phrase only exists in the amended version, not the original. This dependency means Amendment 2 cannot add the public infrastructure criterion without Amendment 1's threshold text already in place, propagating mistakes down the chain.

For illustration, we show a single-step amendment using an example (adapted for brevity) from Civil Code § 31a and a multi-step amendment using an example adapted from Criminal Code § 194.

Single-Step Amendment

Intial Version:

`Volunteer board members whose compensation does not exceed 3,000 euros annually are liable only for intentional or grossly negligent acts.`

Amendment:

`Replace "3,000" with "5,000" to adjust for inflation`

Result:

`Volunteer board members whose compensation does not exceed 5,000 euros annually are liable only for intentional or grossly negligent acts.`

Multi-Step Amendment with Dependencies

Initial version:

`Property damage is prosecuted only if the victim files a criminal complaint.`

Amendment 1:

`After the sentence, insert: "However, damage exceeding 10,000 euros is prosecuted automatically."`

After Amendment 1:

`Property damage is prosecuted only if the victim files a criminal complaint. However, damage exceeding 10,000 euros is prosecuted automatically.`

Amendment 2:

`In the inserted sentence from Amendment 1, replace "10,000 euros" with "10,000 euros or affecting public infrastructure".`

Final result:

Law Amendments Exact Match Rate	Civil Code 177 59.32%	Criminal Code 162 75.93%	Commercial Code 254 52.76%	Income Tax Act 315 30.16%	Overall 908 50.33%
BLEU-1	0.8755 ± 0.2620	0.9515 ± 0.1677	0.9440 ± 0.1385	0.9679 ± 0.0988	**0.9411 ± 0.1658**
BLEU-2	0.8622 ± 0.2891	0.9461 ± 0.1829	0.9385 ± 0.1528	0.9633 ± 0.1014	**0.9344 ± 0.1811**
BLEU-3	0.8566 ± 0.2950	0.9446 ± 0.1870	0.9355 ± 0.1576	0.9589 ± 0.1041	**0.9306 ± 0.1849**
BLEU-4	0.8518 ± 0.2988	0.9429 ± 0.1890	0.9340 ± 0.1603	0.9547 ± 0.1074	**0.9273 ± 0.1875**
ROUGE-1	0.9049 ± 0.2303	0.9551 ± 0.1709	0.9642 ± 0.1163	0.9808 ± 0.0704	**0.9576 ± 0.1456**
ROUGE-2	0.8852 ± 0.2766	0.9499 ± 0.1850	0.9568 ± 0.1403	0.9760 ± 0.0754	**0.9493 ± 0.1696**
ROUGE-L	0.8977 ± 0.2508	0.9533 ± 0.1773	0.9577 ± 0.1289	0.9775 ± 0.0742	**0.9530 ± 0.1567**
BERTScore (P)	0.9573 ± 0.0960	0.9730 ± 0.0972	0.9788 ± 0.0545	0.9895 ± 0.0340	**0.9778 ± 0.0686**
BERTScore (R)	0.9559 ± 0.0994	0.9812 ± 0.0662	0.9763 ± 0.0597	0.9903 ± 0.0288	**0.9785 ± 0.0635**
BERTScore (F1)	0.9561 ± 0.0965	0.9765 ± 0.0841	0.9774 ± 0.0566	0.9898 ± 0.0309	**0.9779 ± 0.0657**

Table 1: Single-step evaluation with mean and standard deviation

Note: Higher values are better. Values show mean ± standard deviation.

```
Property damage is prosecuted only if the victim files a criminal complaint. However, damage exceeding 10,000 euros or affecting public infrastructure is prosecuted automatically.
```

4.1 Data Challenges

Apart from the difficulties introduced by our multi-step setup, the task of automated consolidation presents two challenges. First, PDF extraction is inherently error-prone and yields inaccurate results. Garofalakis et al. (2016) encountered the same problem and achieved only mediocre results compared to Ogawa (2024), who used XML input. In our case, extracting law amendments from Federal Law Gazettes proved particularly problematic because of multi-column formatting and other factors. Second, the absence of an official ground truth dataset tracking all law versions with their in-force intervals forced us to construct one from law gazettes. This process propagated the errors from the PDF extraction, which particularly explains our poor results for the Income Tax Act consolidation, as will be shown below. We therefore view our system not as a mature solution, but as a starting point for further research in German law.

We did reimplement approaches from prior work (see sec. 2) because they target different legal systems and languages with unique amendment conventions, and each handles jurisdiction-specific linguistic patterns. Adapting these to German law amendment formulations would require re-engineering, creating new systems rather than meaningful baselines. Our work establishes the first benchmark for the currently underdeveloped state of automatic German legal consolidation using LLMs.

4.2 Processing and Evaluation

We used OpenAI's GPT-4.1-mini to apply amendments from 148 Federal Law Gazette PDFs to existing laws in XML format, extracting 908 amendments across four legal codes. Three documents exceeded the model's context window, and five documents contained retroactive amendments, where the in-force dates preceded publishing dates. Since the "Laws on the Internet" ground truth is updated daily and does not reflect backdated changes, predictions are compared against outdated versions, making accurate validation impossible. Therefore, eight Federal Law Gazettes were not processed, making a total of 140 processed law gazettes. The problem of law entering into force retroactively, however, cannot be ignored for practical systems and should be revisited in future work.

Using engineered prompts with domain-specific terminology and formatting (see Appendix B), we applied amendments to the initial law versions and compared the generated consolidations with the ground-truth versions. To avoid inflating accuracy with unchanged text, we evaluated only the 908 amended paragraphs, and not the whole law text.

We evaluate consolidation quality using four metrics: two lexical (BLEU and ROUGE) and two semantic (BERTScore). BLEU (Papineni et al., 2002) measures n-gram overlap between predicted and reference texts. We compute BLEU-1 through BLEU-4 with smoothing (Chen and Cherry, 2014) to capture surface-level similarity. However, BLEU cannot detect semantic equivalence. ROUGE (Lin, 2004) complements BLEU by measuring recall through three metrics: ROUGE-1

(unigrams), ROUGE-2 (bigrams), and ROUGE-L (longest common subsequence). To address the limitations of lexical metrics, we also use semantic evaluation. BERTScore (Zhang et al., 2019) uses BERT's contextual embeddings to try to measure semantic similarity, estimating word meanings based on their context.

To complement these automated metrics, a legal expert evaluated 100 randomly selected consolidations that did not exactly match the ground truth. We define an exact match as an exact string match after removing non-printable characters and whitespace. The expert assessed both the severity of the legal impact if the generated text were to be considered law and the effort required to correct discrepancies (see evaluation criteria in Appendix C).

5 Results

The results for the previously tried single-step (5.1) and the novel multi-step (5.2) setup for automated consolidation diverge; while the exact match rate massively deteriorates, the lexical and semantic similarity scores remain high. The expert scores show a mixed picture (5.3).

5.1 Single-Step Amendments

The evaluation covered 908 paragraph-level comparisons from four fundamental German legal codes (Table 1). Exact match rates were 75.93% for Criminal Code, 30.16% for Income Tax Act, and 50.33% overall. The framework achieved 92-95% lexical similarity and 97% semantic similarity across amendments. BERTScore ranged from 99% for the Income Tax Act to 95% for the Civil Code.

BERT-based metrics gave the impression of high semantic equivalence, with BERTScore F1 averaging 0.978 across all codes. Traditional n-gram metrics (BLEU-1 through BLEU-4) showed progressive degradation with longer n-grams, declining from 0.94 to 0.93 overall. ROUGE scores remained high (0.95-0.96).

5.2 Multi-Step Amendment Chains

We also evaluated the framework's capacity to process sequential amendments (Table 2), examining 117 dependency chains with an average length of 2.79 amendments. Due to the increased complexity of the task, multi-step evaluation shows exact match rates declining from 55.56% for Civil Code to just 2.36% for Income Tax Act, with an overall rate of 20.51%.

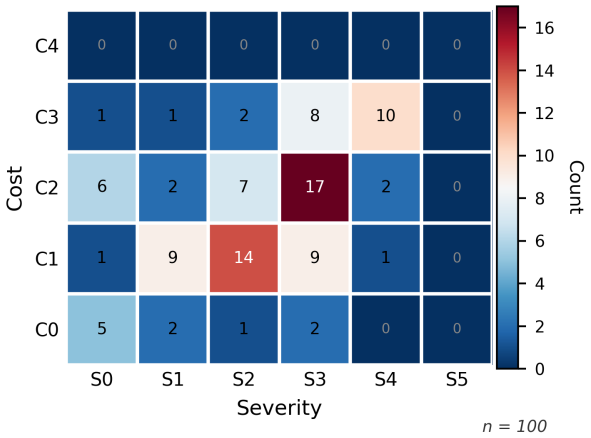

Figure 2: Results of the legal expert evaluation for n = 100 samples with severity of legal impact (x-axis) and correction cost (y-axis)

In lexical and semantic metrics, however, multi-step consolidation nearly maintained performance parity with single-step processing, achieving 97% BERTScore F1, 90.4-87.93% in BLEU 1-4, and 93.9-92.57 in ROUGE 1-L.

5.3 Legal Expert Validation

Figure 2 presents the expert evaluation of 100 randomly sampled imperfect consolidations. The evaluation matrix reveals a concentration of errors with medium and higher severity – with 24% of discrepancies classified as S2 (limited/technical effect) and 36% as S3 (material change within the section) – but relatively low costs of correction – 78% requiring only C0-C2 effort (trivial to single-sentence corrections).

No consolidations exhibited S5 severity errors that would compromise legal validity through constitutional conflicts or clarity violations. 12% showed S4 severity involving rights-critical modifications. 13% and and 14% remained in the low severity sections S0 and S1, respectively. The cost distribution favored minor corrections, with only 22 cases requiring C3-level effort (section-wide redrafts) and none requiring C4 (cross-instrument overhaul).

5.4 Interpretation

When taking the automated metrics and expert evaluation into account, we can establish two contrasting key findings.

Finding 1: From a technical perspective, the 93-99% semantic similarity range across diverse legal domains seems to indicate that LLM-based consolidation preserves meaning with high accuracy. The

Law	**Civil Code**	**Criminal Code**	**Commercial Code**	**Income Tax Act**	**Overall**
Chains	**9**	**28**	**37**	**43**	**117**
Average length	**2.11 ± 0.33**	**2.50 ± 0.92**	**2.70 ± 1.00**	**3.19 ± 1.24**	**2.79 ± 1.09**
Exact Match Rate	**55.56%**	**28.67%**	**27.02%**	**2.36%**	**20.51%**
BLEU-1	0.8749 ± 0.3200	0.9102 ± 0.1769	0.9382 ± 0.1071	0.8850 ± 0.1435	**0.9040 ± 0.1650**
BLEU-2	0.8720 ± 0.3256	0.8971 ± 0.2002	0.9317 ± 0.1100	0.8712 ± 0.1551	**0.8936 ± 0.1781**
BLEU-3	0.8699 ± 0.3269	0.8895 ± 0.2076	0.9274 ± 0.1111	0.8608 ± 0.1613	**0.8861 ± 0.1822**
BLEU-4	0.8683 ± 0.3278	0.8827 ± 0.2132	0.9230 ± 0.1120	0.8520 ± 0.1625	**0.8793 ± 0.1830**
ROUGE-1	0.9058 ± 0.2585	0.9304 ± 0.1641	0.9635 ± 0.0597	0.9325 ± 0.0913	**0.9390 ± 0.1218**
ROUGE-2	0.8851 ± 0.3177	0.9176 ± 0.1891	0.9561 ± 0.0691	0.9175 ± 0.1008	**0.9260 ± 0.1419**
ROUGE-L	0.8943 ± 0.2885	0.9221 ± 0.1783	0.9495 ± 0.0648	0.9174 ± 0.1042	**0.9257 ± 0.1330**
BERTScore (P)	0.9620 ± 0.0966	0.9605 ± 0.0800	0.9794 ± 0.0389	0.9781 ± 0.0396	**0.9734 ± 0.0569**
BERTScore (R)	0.9560 ± 0.1151	0.9698 ± 0.0668	0.9731 ± 0.0468	0.9788 ± 0.0359	**0.9736 ± 0.0551**
BERTScore (F1)	0.9588 ± 0.1111	0.9652 ± 0.0730	0.9761 ± 0.0422	0.9783 ± 0.0376	**0.9734 ± 0.0550**

Table 2: Multi-step evaluation with mean, standard deviation, and chain statistics

Note: Higher values are better. Values show mean ± standard deviation.

framework appears to handle both simple substitutions and complex structural modifications without significant degradation. Also, the equivalence between single-step and multi-step performance in terms of lexical and semantic scores seems to validate the framework's architecture for reconstructing historical law versions through sequential application of amendments. This capability would address a critical gap in Germany's legal infrastructure, as historical versions before 2006 remain unavailable through existing platforms. The ability to process chains of four or more amendments with maintained accuracy would enable the reconstruction of legislative evolution spanning decades.

Finding 2: From a legal perspective, however, the 50.3% and 20.51% rates of exact matches points to the need for extreme caution when working with automatically consolidated statutes. Although the semantic similarity is high in the cases of divergence, such metrics are misleading. The legal language deviates from everyday language in the sense that it uses terms with clearly defined meanings, which cannot be exchanged with synonyms and are often detached from ordinary meaning – this has been shown to be true for German legal language in particular Behnke and Wais (2023). Here, the expert evaluation provides crucial context for the automated metrics. Most of the mistakes were found to change the meaning of a statute and thus create room for legal uncertainty or misinterpretation.

On the positive side, the costs of adjustments were overall rated to be manageable – it should not be overlooked, however, that amendments usually introduce little change and mishandled consolidation will thus in general lead to errors that are easy to fix. Also, the absence of validity-threatening errors (S5) and the minimal occurrence of rights-critical changes (S4) seem to indicate that the framework's failure modes are bounded. Yet, one has to take into account that the severity of errors is heavily influenced by the nature of the statute affected; errors in provisions of criminal law, for example, will generally be considered to be more rights-critical than civil law provisions – in our experiments, the LLM performed best on the former in terms of exact matches, but this might not be the case for other legal systems. The differences in the ratio of exact matches between the different legal areas point to the hypothesis that the system's error rate increases with the complexity of provisions, which is low in criminal law and high in business law Katz et al. (2020) and the very technical tax law.

6 Prototype for Experimentation

We developed a prototype web application for consolidating German federal laws, planned for public experimentation. The interface (Figure 3) provides a three-step workflow: users select a law (Civil Code, Criminal Code, Commercial Code, or Income Tax Act), the system processes it through twelve automated steps (10-30 minutes depending on complexity), and users access all versions with timestamps and validity status. The prototype creates version histories using enforcement dates extracted during amendment processing (Section 4). Each version is marked as historical or currently valid. Currently, the system reconstructs versions from 2019 onward, but the framework can extend to amendments published since 1949. Users can

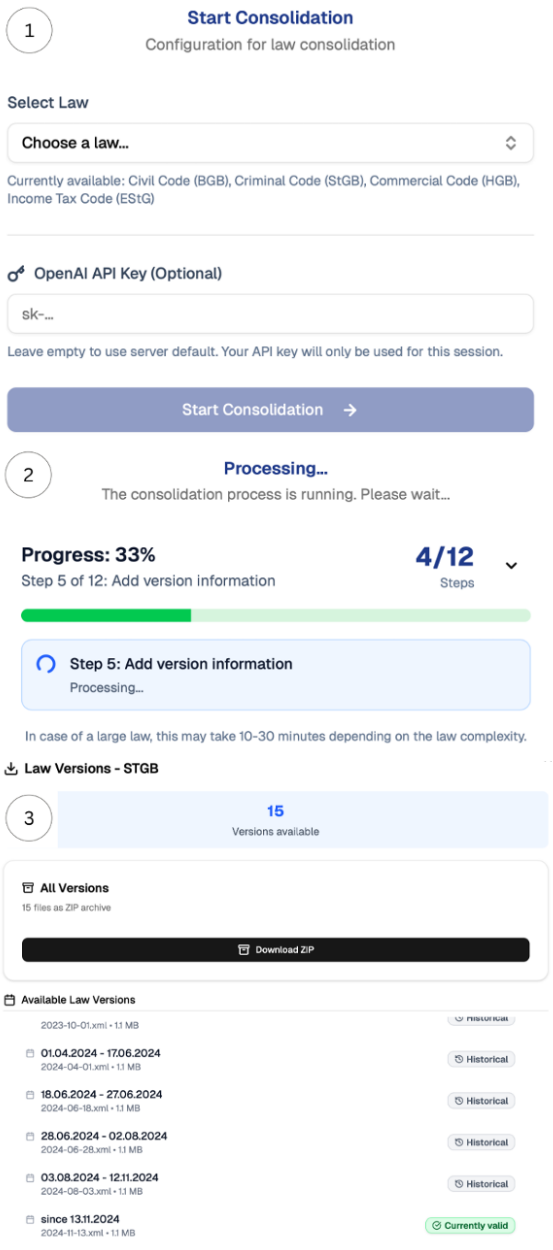

Figure 3: User Interface of our prototype.

download any version as an archive.

7 Conclusion and Future Work

This research presents the first investigation into an LLM-based approach to consolidating German statutes, examining the practical viability of addressing the task with technical means. Our experimental framework successfully processed 140 of 148 Federal Law Gazette documents (94.5%) and achieved 93-99% semantic similarity even with complex amendment chains averaging 2.79 amendments. However, the exact match rates remained very low (50.3% for single-step and 20.51% for multi-step). Given the peculiarities of the legal language with its strict definitions of technical terms, we pointed out that these low rates make human evaluation paramount. While our own expert evaluation of 100 imperfect consolidations revealed that 53% of discrepancies were cosmetic or had limited technical effects (S0-S2), with 78% requiring only trivial corrections (C0-C2) and no validity-threatening errors, relying on lexical or semantic scores in the cases of non exact matches alone would severely overestimate the performance of such systems. We therefore recommend strict oversight when using LLMs for the task of automated consolidation.

Three documents exceeded the model's context window, revealing a critical limitation. Future work should develop chunking strategies for lengthy legal documents that preserve semantic relationships within context constraints, enabling processing of currently inaccessible documents and improving existing consolidations. Another direction involves an agentic framework that dynamically selects models based on amendment complexity. Simple substitutions would use smaller, cost-effective models, while complex amendments with cross-references or dependency chains would trigger larger models. This adaptive approach optimizes the cost-accuracy trade-off, making large-scale deployment economically feasible while maintaining quality for critical consolidations. These improvements could enable comprehensive automation of German federal legal consolidation, transforming legal accessibility for practitioners, courts, and citizens.

References

Timothy Arnold-Moore. 1995. Automatically processing amendments to legislation. In *Proceedings of the 5th International Conference on Artificial Intelligence and Law (ICAIL)*, pages 297–306. ACM.

Timothy Arnold-Moore. 1997. Automatic generation of amendment legislation. In *Proceedings of the 6th International Conference on Artificial Intelligence and Law (ICAIL)*, page 56–62. ACM.

Gregor Behnke and Niklas Wais. 2023. On the semantic difference of judicial and standard language. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law (ICAIL 2023)*, pages 382–386, Braga, Portugal. ACM.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level BLEU. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367, Baltimore, Maryland, USA. Association for Computational Linguistics.

Matias Etcheverry, Thibaud Real, and Pauline Chavallard. 2024. Algorithm for automatic legislative text consolidation. In *Proceedings of the Natural Legal Language Processing Workshop 2024*, pages 166–175, Miami, FL, USA. Association for Computational Linguistics.

Samuel Fabrizi, Maria Iacono, Andrea Tesei, and Lorenzo De Mattei. 2021. A first step towards automatic consolidation of legal acts: Reliable classification of textual modifications. In *Proceedings of the Workshop on Technologies for Regulatory Compliance (TechReg 2021)*. CEUR-WS. Available under Creative Commons Attribution 4.0 International (CC BY 4.0).

John Garofalakis, Konstantinos Plessas, and Athanasios Plessas. 2016. A semi-automatic system for the consolidation of greek legislative texts. In *Proceedings of the 20th Pan-Hellenic Conference on Informatics (PCI 2016)*. ACM.

Daniel Martin Katz, Corinna Coupette, Janis Beckedorf, and Dirk Hartung. 2020. Complex societies and the growth of the law. *Scientific Reports*, 10(1):18737.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Masataka Ogawa. 2024. Syntactic cues may not aid human parsers efficiently in predicting Japanese passives. In *Proceedings of the 38th Pacific Asia Conference on Language, Information and Computation*, pages 1490–1503, Tokyo, Japan. Tokyo University of Foreign Studies.

Yasuhiro Ogawa, Shintaro Inagaki, and Katsuhiko Toyama. 2008. Automatic consolidation of japanese statutes based on formalization of amendment sentences. In *New Frontiers in Artificial Intelligence: JSAI 2007 Conference and Workshops, Miyazaki, Japan, June 18–22 2007, Revised Selected Papers*, volume 4914 of *Lecture Notes in Computer Science*, pages 363–376. Springer.

Monica Palmirani and Fabio Vitali. 2012. Legislative xml: Principles and technical tools.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Limitations

Our approach faces several technical and legal constraints. Three documents exceeded GPT-4.1-mini's context window, preventing processing of lengthy amendments. PDF extraction from Federal Law Gazettes introduced errors that propagated through the pipeline, particularly affecting the Income Tax Act results. The absence of official ground truth for historical law versions required constructing our own dataset, limiting validation accuracy. Most critically only 50.3% of single step consolidations and 20.5% of multi-step consolidations matched exactly—a crucial limitation since legal language requires precise terminology where synonyms can alter legal meaning. The system cannot process retroactive amendments, so all outputs must be treated as drafts requiring expert review.

Ethical Statement

We used generative AI for code and drafting. The dataset includes only publicly available Federal Law Gazettes and official consolidated laws; no personal data was processed. Automated consolidation can broaden access but must not replace authoritative sources. Errors could affect rights and obligations, so deployments should include clear disclaimers and expert review, especially for criminal law. The framework is a support tool for professionals, not an autonomous legal authority.

Acknowledgements

This work was carried out within the project "Generatives Sprachmodell der Justiz (GSJ)", a joint initiative of the Ministry of Justice of North Rhine-Westphalia (Ministerium der Justiz des Landes Nordrhein-Westfalen) and the Bavarian State Ministry of Justice (Bayerisches Staatsministerium der Justiz), with the scientific partners Technical University of Munich (Technische Universität München) and University of Cologne (Universität zu Köln). The project is financed through the Digitalisierungsinitiative des Bundes für die Justiz.

Appendix

A Prompt: Extract amendment from law gazette

System Message:

```
You are an expert in German law and JSON formatting.
ABSOLUTELY CRITICAL:
1. You MUST extract the COMPLETE content of each article
2. NEVER shorten, summarize, or omit
3. If an article has 1000 lines, copy ALL 1000 lines
4. Phrases like "text as above" or "..." are STRICTLY FORBIDDEN
EFFECTIVE DATE NEVER NULL!
- There is ALWAYS effective date information in the legal text
- Search mandatorily for the last article about "Entry into Force"
- standard_inkrafttreten must NEVER be empty or null!
You MUST return valid JSON with correct escaping.
```

Main Extraction Prompt:

```
Analyze the following German law and extract all articles structurally.
ABSOLUTE CRITICAL RULE - NEVER SHORTEN!
YOUR TASK:
1. Find ALL articles in the text (begin with "Article" followed by a number)
2. Extract for each article:
- Number (only the digit after "Article")
- Title (text directly after "Article X")
- Complete content (ABSOLUTELY EVERYTHING - NO SHORTCUTS!)
- Which law is amended (search for phrases like "is amended as follows")
- All amended paragraphs (EACH paragraph must begin with §)
3. Find the effective date (usually in the last article "Entry into Force")
4. Extract the effective date rules in detail - THERE MUST ALWAYS BE A DATE!
```

FNA Assignment (when -fna parameter specified):

```
You are executed with -fna {target_fna}, which means you should assign
articles that amend the following law: {fna_info}
IMPORTANT ASSIGNMENT RULES:
1. Check EXACTLY the title of each article - it states which law is amended
2. If article EXPLICITLY amends target law → zugeordnete_fna = "{target_fna}"
3. If article amends ANOTHER law → zugeordnete_fna = null
4. BUT: Ensure AT LEAST ONE article is assigned to FNA {target_fna}
5. If uncertain, assign the MOST LIKELY article
6. ONLY ONE uncertain article gets assigned - further uncertain articles get null
```

Output Format:

```
ANSWER AS VALID JSON (COMPLETE CONTENT - NO SHORTCUTS!):
{
"standard_inkrafttreten": "YYYY-MM-DDTHH:MM:SS+01:00",
"inkrafttreten_regeln": [...],
"artikel": [{
"nummer": "X",
"titel": "Title of article",
"inhalt": "COMPLETE TEXT - EVERYTHING! EVERY LETTER!",
```

```
"geaendertes_gesetz_name": "Name of amended law",
"zugeordnete_fna": "XXX-X or null",
"geaenderte_paragraphen": ["§ X", "§ Y"]
}]}
```

Parameters:

- Model: GPT-4.1
- Temperature: 0
- Response format: JSON object
- Variables: `{year}` = document year, `{pdf_text}` = preprocessed PDF content, `{target_fna}` = optional FNA filter, `{fna_info}` = list of FNA codes with law names

B Prompt: Apply amendment to initial version

System Message (for all prompts):

```
You are a precise legal text processor for {JURABK}. Preserve all existing structure exactly while making only necessary changes.
```

Prompt 1: Modifying Existing Legal Text

```
You are a legal text processor. You need to apply the legal change description to the XML content for {PARAGRAPH}.
Use the provided XML as your COMPLETE GUIDELINE and template. Preserve all existing structure exactly while making necessary changes.
Original XML content (use as complete guideline):
{ORIGINAL_XML}
Legal change description:
{CHANGE_CONTENT}
CRITICAL REQUIREMENTS:
1. **USE INITIAL FILE AS COMPLETE GUIDELINE**: Follow the exact structure, formatting, and style shown in the original XML above
2. **PRESERVE ALL EXISTING ELEMENTS**: Keep all existing XML tags, attributes, indentation, and formatting exactly as they are
3. **PRESERVE METADATA**: Keep builddate, doknr, jurabk, enbez, titel exactly as shown in the original
4. **ALLOW NECESSARY ADDITIONS**: You may ADD new XML elements when required by the legal changes
5. **MAINTAIN CONSISTENT STYLE**: Any new elements must match the indentation and formatting style
Return the complete modified XML:
```

Prompt 2: Creating New Legal Paragraphs

```
You are a legal text processor. Create the complete XML content for NEW legal paragraph {PARAGRAPH}.
Legal change description:
{CHANGE_CONTENT}
Use this template and follow the exact formatting:
{XML_TEMPLATE}
```

`Return the complete XML with proper formatting and indentation:`

Parameters:

- Model: GPT-4.1-mini
- Temperature: 0.1
- Variables: `{JURABK}` = legal code (e.g., BGB, StGB), `{PARAGRAPH}` = section number, `{ORIGINAL_XML}` = current law XML, `{CHANGE_CONTENT}` = amendment text, `{XML_TEMPLATE}` = structure for new paragraphs

C Legal expert evaluation

Severity S0–S5 (legal impact if model text were law). Choose the highest fitting level.

S0 Cosmetic only (spelling/punctuation/layout).

S1 Minimal debate risk; meaning effectively unchanged.

S2 Limited/technical effect (minor content or cross-reference; similar outcome likely).

S3 Material change within the Section (Tatbestand (legal elements), thresholds, exceptions, addressees, Legal consequence (German: Rechtsfolge)).

S4 Major or rights-critical change ("may" (German: kann/darf) / "should" (German: soll) / "must" (German: muss); "and"/"or" (German: und/oder); sanctions/competence; broad scope).

S5 Critical/validity risk (Basic Law (German: Grundgesetz)/EU conflict; Legal clarity requirement (German: Bestimmtheit).

Cost C0–C4 (effort to align to ground truth; not the legal impact).

C0 Trivial patch (single token/punctuation).

C1 Single-sentence edit; no propagation.

C2 Local multi-sentence/structure fix; local renumber/cross-reference.

C3 Section-wide redraft or propagated references/definitions across the legal provision.

C4 Cross-instrument/systemic overhaul (impacts regulations (German: Verordnungen), annexes, sanction scales).