RegTrack: A Fine-Grained Benchmark for Multi-Class Legal Change Detection

Joe Yu* **Kevin Chenhao Li*** **Julian Ostarek**
Technical University of Munich
{joe.yu, kevinchenhao.li, julian.ostarek}@tum.de

Abstract

Organizations must continuously monitor evolving regulations to maintain compliance. While current tools are limited to surface-level text comparison, existing models lack the fine-grained classification schemes to determine whether small changes impact legal obligations or merely update formatting. To address this gap, we introduce a novel benchmark for change detection in EU regulations. It comprises 4,772 manually annotated pairs of structurally distinct provisions, defined as Atomic Legal Units (ALUs), mapped to a six-class taxonomy of legal change types. We formalize three core tasks: structural alignment, change classification, and a combined task requiring simultaneous alignment and classification. Evaluating lexical algorithms, dense encoders, and Large Language Models (LLMs) as baselines, we find LLMs excel at isolated change classification, whereas domain-specific dense encoders offer the most robust combined performance. By providing fine-grained labeled data, this benchmark enables the development of AI systems that can help organizations analyze regulatory shifts and support version-aware retrieval in the legal domain.

1 Introduction

Legal regulations evolve continuously to address societal needs, technological advances, and policy shifts. Understanding these changes is crucial for legal practitioners, policymakers, and organizations that must ensure compliance with current regulatory requirements (Katz et al., 2020). However, manually tracking and analyzing amendments across thousands of legal documents is time-intensive, error-prone, and requires significant domain expertise (Breton et al., 2025; Fürst et al., 2025).

Legal change detection extends beyond simple text comparison, as a single altered word can fundamentally shift a provision's meaning (Côté, 1991; Alschner, 2020). Automated systems must therefore classify whether a modification meaningfully impacts legal obligations or merely updates formatting, such as modernizing grammar or fixing typos. Because amendments also encompass reference updates, structural reorganizations, and provision additions or deletions, distinguishing these change types is essential for assessing compliance impact, prioritizing review, and determining when action is required from organizations.

While recent advances in Natural Language Processing (NLP) have demonstrated promising results in high-level legal text analysis tasks (including named entity recognition, document classification, and semantic similarity detection (Ariai et al., 2025; Siino et al., 2025)), the application of NLP to automated legal change detection remains underexplored. This is particularly true for the fine-grained classification of amendment types in multi-version legal corpora.

To address this gap, we introduce a novel dataset for EU Regulation Change Detection and Classification. It comprises 4,772 manually annotated pairs of legal text segments from 6 EU regulations across multiple consolidated versions, spanning the environmental, transportation, and emissions control domains. Each pair represents an alignment between Atomic Legal Units (ALUs), which are structurally distinct provisions, such as articles, paragraphs, or numbered points, serving as the smallest semantically complete segments of legal text. These alignments between consecutive regulation versions are labeled with one of six expert-validated change types: no change, syntactic change, reference update, semantic change, addition, and deletion.

2 Related Work

Legal NLP & Change Detection. While Legal NLP has advanced rapidly (Chalkidis et al.,

*Equal contribution

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 764–778

2020; Henderson et al., 2022), most work focuses on case law or high-level document classification (Chalkidis et al., 2021), leaving fine-grained legislative change detection underexplored. Existing solutions often rely on proprietary financial data (Abualhaija et al., 2024) or purely lexical differencing algorithms. For instance, DocuToads (Hermansson and Cross, 2016) improves standard edit-distance (Lcvenshtcin, 1966) by successfully handling text transpositions in statutes. However, as noted by Alschner (2020), these lexical tools lack the semantic awareness to distinguish trivial reformatting from substantive legal shifts. Closely related to our work, Li et al. (2022) distinguish "stylistic" from "relevant" treaty differences. Yet, their approach relies on noisy, distantly supervised labels and a coarse three-class taxonomy, whereas our task requires expert-validated, fine-grained annotations.

Version-Aware Retrieval. Moving beyond isolated change detection, temporal Retrieval-Augmented Generation systems focus on navigating document evolution over time (e.g., VersionRAG (Huwiler et al., 2025), SAT-Graph RAG (de Martim, 2025b)). While corpora like CLAW (Xu et al., 2025) enable historical search and frameworks like LRMoo (de Martim, 2025a) model legal history, a distinct gap remains. These systems retrieve past versions but lack the granular classifications to map exactly *how* the text evolved. By providing expert annotations anchored to stable ALUs, our dataset supplies the semantic links needed to power truly version-aware legal retrieval.

3 Methodology

Our methodology for constructing the dataset consists of three primary phases: (1) data collection from EUR-Lex, (2) preprocessing and automated alignment, and (3) manual annotation and verification. This pipeline transforms raw regulatory HTML documents into annotated training data suitable for supervised machine learning.

3.1 Data Collection

All regulatory texts are sourced from EUR-Lex (European Union, 2026), the official database of European Union law. EUR-Lex provides consolidated versions of regulations, which incorporate all amendments and corrigenda up to a specific date, enabling systematic comparison between regulatory states at different points in time.

After identifying regulations and directives with sufficient version history for temporal comparison, we queried the SPARQL endpoint (Publications Office of the European Union, 2026a) using the CELEX identifier for all consolidated versions in the form of XML files. The data retrieval was performed on 23 December 2025.

3.2 Preprocessing and Alignment

The structure of EU legal acts is defined by the Interinstitutional Style Guide (Publications Office of the European Union, 2026b) and the Joint Practical Guide (European Parliament and Council of the European Union and European Commission, 2015). The primary numbered subdivisions of these acts are Articles, Paragraphs, and Points. We used these official subdivisions as the basis for our document segmentation, which allowed us to split the text into semantically complete units while tracking changes at a fine-grained level.

For each consecutive version pair, we parsed both texts into hierarchical ALUs with stable IDs and aligned them in three passes: (i) exact ALU-ID matching, (ii) text-similarity-based matching for unmatched ALUs, and (iii) labeling remaining unmatched ALUs as `addition` or `deletion`. For each aligned pair, we then assigned an initial label using rule-based heuristics: exact matches were labeled `no_change`, citation-only updates were labeled `reference_update`, and changes in key numbers, dates, or units (e.g., g/km) were labeled `semantic_change`. Among the remaining aligned pairs, high-similarity edits were labeled `syntactic_change`, and all others were labeled `semantic_change`. This automated classification was not used to directly create the final gold labels. Rather, it served as a pre-annotation step to scaffold and speed up the manual annotation process.

3.3 Annotation

To align interpretations of the labeling guidelines (see Appendix A), the three annotators, who are computer science researchers actively working on legal NLP projects, participated in a shared calibration session. A representative subset of the regulation data was labeled independently, followed by a collective review of disagreements. This led to a refinement of the Atomic Legal Unit boundaries and change class definitions.

Following calibration, the remaining dataset was partitioned, and each ALU was assigned to two primary annotators. During this phase, annotators were presented with the automated structural align-

ments and the initial rule-based classification labels. They were tasked with manually verifying both the structural alignment and the change classification. To prevent interpersonal influence among the human reviewers, the annotators worked independently without access to each other's decisions. While seeing the initial rule-based suggestions may have introduced some anchoring effect, the gold labels were determined solely by human agreement. A third annotator acted as a mediator in cases of disagreement. Final labels were determined by accepting the label when the two primary annotators agreed and by deferring to the third annotator's decision when they disagreed.

To assess annotation reliability, we calculated inter-annotator agreement. We achieved a simple agreement percentage of 0.9707 for the structural alignment verification and a Cohen's κ of 0.9332 for the change classification, indicating very high agreement across both tasks. Labeling the data according to the structure laid out in Appendix A.2 establishes a dominance hierarchy of change.

4 Dataset Analysis

Our dataset consists of 6 regulations across 18 consolidated versions with a total of 4,772 manually annotated entries.

4.1 Class Distribution

Label	Total	Percentage
No_change	3,545	74.29%
Syntactic_change	71	1.49%
Reference_update	10	0.21%
Semantic_change	266	5.57%
Addition	677	14.19%
Deletion	203	4.25%

Table 1: Distribution of labels across the ALU pairs in the dataset.

As shown in Table 1, the dataset is dominated by unchanged entries, reflecting the granular way in which amendments and corrigenda target specific provisions while leaving surrounding regulatory text unchanged (European Parliament and Council of the European Union and European Commission, 2015). Additions are more frequent than deletions, consistent with the tendency of legal texts to grow rather than shrink over time (Katz et al., 2020). The prevalence of semantic over syntactic changes is also consistent with the view that amendments are made "to change its substance and not simply to improve its written expression" as quoted in Erasmus (1998).

As an auxiliary analysis of cross-lingual consistency, we used the manually annotated English labels as ground truth to infer corresponding labels for other language versions. Leveraging the multilingual nature of EU law and the principle that the same provision should retain its meaning across languages, we applied the dominance hierarchy and inference rules (see Appendix F) to the German manifestations and validated the inferred labels manually. This auxiliary analysis assessed cross-lingual label transfer only. All reported benchmark experiments were conducted on the English dataset.

5 Experiments

The dataset structure allows us to define multiple change detection tasks: alignment, classification, and a combination of both.

The alignment task pairs a query ALU with its matching target ALU from the full set of ALUs in the opposing version of the same regulation version pair, without truncation or additional prefiltering. This full-pool search inherently captures additions and deletions: queries without a match are classified as a deletion in forward search and as an addition in backward search.

The classification task determines the change type for an already aligned ALU pair. Since additions and deletions involve only a single ALU, they are excluded from this task.

The combined task reflects a realistic end-to-end setting. In practice, a system is not given perfectly aligned legal provisions. It must first find the corresponding provision across versions and then classify the change. For this task, a prediction is considered correct only if both the predicted counterpart ALU is correct (or NONE is correctly predicted for additions and deletions) and the predicted change label is correct. If a model chooses the wrong ALU, the combined prediction is counted as incorrect regardless of the predicted label. This evaluation demonstrates whether a method works for the full legal change detection pipeline rather than just the isolated classification case.

Due to class imbalance, we report Macro-F1 for the classification-only and combined tasks.

Model Name	Alignment	Classification	Combined Task	
	Average accuracy	Macro F1	Macro F1	Accuracy
Lexical & Sparse				
Levenshtein Ratio	0.7000	**0.5556**	0.5250	0.8961
DocuToads	**0.7808**	0.5444	**0.5477**	**0.9074**
BM25	0.3333	0.3910	0.2009	0.3029
splade	0.7790	0.5480	0.4755	0.8961
Dense Encoders				
bert-base-uncased	0.6053	0.6853	0.4748	0.7660
nlpaueb/bert-base-uncased-eurlex	0.4934	0.6689	0.4377	0.8924
all-MiniLM-L6-v2	0.7959	0.6650	0.5967	**0.9186**
all-MiniLM-L12-v2	0.7887	0.6818	0.5652	0.9099
paraphrase-mpnet-base-v2	**0.7965**	0.6657	0.5757	0.9011
paraphrase-multilingual-mpnet-base-v2	0.7923	**0.7091**	**0.6001**	0.9086
Qwen3-Embedding-0.6B	0.7814	0.6432	0.5343	0.8636
Qwen3-Embedding-4B	0.7814	0.6432	0.5073	0.8636
Generative LLMs				
Llama3-8b-instruct	0.3720	0.7148	0.3536	0.5257
DeepSeek-V3.2	**0.7084**	**0.9158**	**0.5675**	**0.8698**
Llama-4-Maverick-17B-128E-Instruct	0.4974	0.8685	0.3783	0.5957
gpt-oss-120b	0.6263	0.8574	0.5318	0.7685

Table 2: Task-wise and combined test set performance for threshold tuned models. Alignment average accuracy denotes the unweighted mean of per-class alignment accuracies, where an alignment is correct if the model selects the gold counterpart ALU or correctly predicts NONE for additions and deletions. Best results in each category are marked in bold.

5.1 Baselines

We examined three families of baselines: lexical methods, dense encoders, and generative Large Language Models (LLMs). These baselines cover three increasingly semantic families of methods that are natural for this task and common in related work. Lexical baselines include the BM25 sparse retrieval function, DocuToads (Hermansson and Cross, 2016), and normalized Levenshtein similarity (Bachmann, 2024). For dense encoders, we evaluated standard and legal-domain fine-tuned BERT models (Devlin et al., 2019; Chalkidis et al., 2020), applying Whitening (Su et al., 2021; Li et al., 2020) to combat anisotropy. We also tested other popular dense and sparse embedding architectures, such as Sentence-BERT, Qwen3, and SPLADE (Reimers and Gurevych, 2019; Zhang et al., 2025; Formal et al., 2021). Lastly, we evaluated a diverse set of generative LLMs, including models from the Llama family (Grattafiori et al., 2024), DeepSeek-V3.2 (DeepSeek-AI et al., 2025), and gpt-oss-120b (OpenAI et al., 2025), using fixed prompt templates (see Appendix G).

For all lexical, sparse, and dense non-generative baselines, the same pairwise scoring function was used for both alignment and classification. In the alignment task, scores were used to rank candidate ALUs. In the classification-only task, the score of the given old and new ALU pair was mapped to a change label using thresholds tuned on the training split. Importantly, no model weights were fine-tuned during our experiments. All LLMs and dense encoders were evaluated in a zero-shot inference setting. Full details regarding inference hyperparameters, including LLM decoding temperatures and threshold search ranges, are provided in Appendices G and H.

The complete performance results across all tasks are summarized in Table 2. The results reveal a stark contrast between isolated and end-to-end performance. Generative LLMs, particularly DeepSeek-V3.2, dominate the isolated classifica-

tion task with a Macro F1 of 91.6%. However, their performance degrades significantly in the combined setting (56.8%) due to poorer structural alignment capabilities. Conversely, domain-specific dense encoders offer the most robust end-to-end solutions, with `paraphrase-multilingual-mpnet-base-v2` achieving the highest combined Macro F1 (60.0%) and `all-MiniLM-L6-v2` reaching the highest combined accuracy (91.9%). Traditional lexical methods like BM25 struggle on semantic distinctions, though specialized algorithms like DocuToads remain highly competitive in isolated alignment (78.1%). More detailed per-class metrics can be found in Appendix I.

To illustrate common failure modes of existing semantic change detection systems, we include one likely false positive and one likely false negative with full ALU text in Appendix B.

6 Conclusion

In this work, we introduced a novel, expert-annotated benchmark for fine-grained change detection in EU regulations. By grounding the dataset in ALUs and a six-class semantic taxonomy, we move beyond simple lexical differencing toward a more precise characterization of how legislative text evolves. Our baseline evaluations show that generative LLMs perform best on isolated change classification, while domain-specific dense embedding models provide the strongest end-to-end baseline on the combined task. At the same time, the performance drop observed across models when alignment and classification are coupled shows that fine-grained legal change detection remains an open challenge. We hope this dataset will support future work on version-aware legal retrieval, temporal reasoning, and automated assessment of regulatory change.

Limitations

While our dataset provides a novel benchmark for fine-grained regulatory change detection, it has several limitations. Firstly, the data is restricted to EU regulations within specific domains. Future work could evaluate how our baselines generalize to other jurisdictions, such as common law, or different document types like case law. Secondly, our parsing pipeline excludes annexes and appendices, leaving the extraction of their complex tables and figures as an open challenge for future research. Thirdly, the dataset exhibits a severe class imbalance, with unchanged text dominating while specific modifications remain rare. Although this reflects true legislative distributions, it complicates evaluation, as aggregate metrics can obscure a model's true performance on rare but critical change types. Future work could address this by curating balanced evaluation subsets or developing specialized metrics to better assess these minority classes. Finally, because EUR-Lex is a public database, the evaluated LLMs may have been exposed to parts of the underlying source texts during pre-training. While our alignment tasks and taxonomy are entirely novel, this potential data contamination could artificially inflate zero-shot and few-shot performance.

References

Sallam Abualhaija, Marcello Ceci, Nicolas Sannier, Domenico Bianculli, Lionel C. Briand, Dirk Zetzsche, and Marco Bodellini. 2024. Ai-enabled regulatory change analysis of legal requirements. In *2024 IEEE 32nd International Requirements Engineering Conference (RE)*, pages 5–17.

Wolfgang Alschner. 2020. Sense and similarity: automating legal text comparison. In *Computational Legal Studies*, pages 9–28. Edward Elgar Publishing.

Farid Ariai, Joel Mackenzie, and Gianluca Demartini. 2025. Natural language processing for the legal domain: A survey of tasks, datasets, models, and challenges. *ACM Computing Surveys*, 58(6):1–37.

Max Bachmann. 2024. rapidfuzz/rapidfuzz: Release 3.6.2.

Julien Breton, Mokhtar Mokhtar Billami, Max Chevalier, Ha Thanh Nguyen, Ken Satoh, Cassia Trojahn, and May Myo Zin. 2025. Leveraging llms for legal terms extraction with limited annotated data. *Artificial Intelligence and Law*, pages 1–27.

Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. 2021. MultiEURLEX - a multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6974–6996, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Pierre-André Côté. 1991. *The interpretation of legislation in Canada*. Cownsville, Qu ebec: Editions Y. Blais.

Hudson de Martim. 2025a. Modeling the diachronic evolution of legal norms: An lrmoo-based, component-level, event-centric approach to legal knowledge graphs. *arXiv preprint arXiv:2506.07853*.

Hudson de Martim. 2025b. An ontology-driven graph rag for legal norms: A structural, temporal, and deterministic approach. *arXiv preprint arXiv:2505.00039*.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, and 245 others. 2025. Deepseek-v3.2: Pushing the frontier of open large language models. *Preprint*, arXiv:2512.02556.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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.

Janet Erasmus. 1998. Plain language drafting meets interpretive principles and rules–a drafter's perspective. Canadian Institute for the Administration of Justice.

European Parliament and Council of the European Union and European Commission. 2015. *Joint practical guide for persons involved in the drafting of European Union legislation*. Publications Office of the European Union.

European Union. 2026. EUR-Lex: Access to European Union law. Accessed: 2026-02-23.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2288–2292, New York, NY, USA. Association for Computing Machinery.

Daniel Fürst, Mennatallah El-Assady, Daniel A Keim, and Maximilian T Fischer. 2025. Challenges and opportunities for visual analytics in jurisprudence. *Artificial Intelligence and Law*, pages 1–32.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Peter Henderson, Mark Krass, Lucia Zheng, Neel Guha, Christopher D Manning, Dan Jurafsky, and Daniel Ho. 2022. Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset. In *Advances in Neural Information Processing Systems*, volume 35, pages 29217–29234. Curran Associates, Inc.

Henrik Hermansson and James P Cross. 2016. Tracking amendments to legislation and other political texts with a novel minimum-edit-distance algorithm: Docutoads. *arXiv preprint arXiv:1608.06459*.

Daniel Huwiler, Kurt Stockinger, and Jonathan Fürst. 2025. Versionrag: Version-aware retrieval-augmented generation for evolving documents. *arXiv preprint arXiv:2510.08109*.

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

VI Lcvenshtcin. 1966. Binary coors capable or 'correcting deletions, insertions, and reversals. In *Soviet physics-doklady*, volume 10.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Xiang Li, Jiaxun Gao, Diana Inkpen, and Wolfgang Alschner. 2022. Detecting relevant differences between similar legal texts. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 256–264, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, and 108 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Publications Office of the European Union. 2026a. Cellar: The EU common content and metadata repository and SPARQL endpoint. `https://publications.europa.eu/en/web/about-us/legal-notices/cellar`. Data retrieved via SPARQL endpoint: `https://publications.europa.eu/webapi/rdf/sparql`.

Publications Office of the European Union. 2026b. Interinstitutional style guide. Accessed: February 23, 2026.

Python Software Foundation. 2026. difflib — helpers for computing deltas. `https://docs.python.org/3/library/difflib.html#difflib.SequenceMatcher`. Accessed: 2026-02-24.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Marco Siino, Mariana Falco, Daniele Croce, and Paolo Rosso. 2025. Exploring llms applications in law: A literature review on current legal nlp approaches. *IEEE Access*, 13:18253–18276.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Xinzhe Xu, Liang Zhao, Hongshen Xu, and Chen Chen. 2025. Claw: Benchmarking chinese legal knowledge in large language models-a fine-grained corpus and reasoning analysis. *arXiv preprint arXiv:2509.21208*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, and 1 others. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

A Labeling Guideline

This appendix defines the taxonomy for classifying changes between regulatory versions at the **Atomic Legal Unit (ALU)** level, which is the smallest independent paragraph or subparagraph.

A.1 Level 1: Textual Detection

`No_change` 100% identical text strings.

`Change` Any character-level divergence (punctuation, spelling, words).

A.2 Level 2: Nature of Change

Applied only if Level 1 is identified as a *Change*.

Syntactic_change/Rewording The text changed, but the legal meaning is identical.

- Includes: Modernizing grammar, fixing typos, or changing reference styles.
- Example: "Article 10" to "Decision XYZ."

Semantic_change/Modification The legal meaning, which includes but isn't limited to: the obligation, target, metric, or scope has changed.

- Example: from "37.5%" to "55%."

Reference_update Reserved for exchanging references where the internal ALU text remains stable. If the new reference alters the obligation, it is a *Modification*.

Addition A completely new provision/paragraph that didn't exist before

Deletion An existing ALU removed from the new version.

A.3 The Compliance Impact Rule

> *If a manufacturer must alter their behavior or reporting due to this change, it is a **Modification**. If they can continue existing operations exactly as before, it is a **Rewording**.*

B Representative Error Cases for Existing Systems

This appendix provides two representative ALU pairs where existing semantic change detection systems are likely to fail: one false positive (semantic over-prediction) and one false negative (semantic under-prediction).

B.1 False Positive: Lexically Different but Semantically Stable

Field	Value
CELEX ID	32003L0087
ALU ID	`Art10_Par4_Suba`
Versions	2017-12-29 → 2018-04-08
Gold Label	`syntactic_change`
Old text: *operators, and in particular any SMEs covered by the Community scheme, have full, fair and equitable access;*	
New text: *operators, and in particular any small and medium-sized enterprises covered by the EU ETS, have full, fair and equitable access;*	

Table 3: False Positive Error Case: Lexically Different but Semantically Stable

Observation: Term expansion ("SMEs" → "small and medium-sized enterprises") and terminology modernization ("Community scheme" → "EU ETS") can trigger semantic detectors, although the operative requirement is unchanged. This demonstrates a scenario where traditional lexical baselines often fail, but generative LLMs are generally better equipped to predict a syntactic change.

B.2 False Negative: High Lexical Overlap but Scope Expansion

Observation: The sentence remains highly similar, but the legal scope is expanded by adding *shipping company* as an additional obligated actor. Similarity-heavy systems may under-predict this as non-semantic. This highlights a case where dense encoders averaging token embeddings often

Field	Value
CELEX ID	32003L0087
ALU ID	`Art12_Par2`
Versions	2023-03-01 → 2023-06-05
Gold Label	`semantic_change`

Old text:
Member States shall ensure that allowances issued by a competent authority of another Member State are recognised for the purpose of meeting an aircraft operator's obligations under paragraph 2a or of meeting an operator's obligations under paragraph 3.

New text:
Member States shall ensure that allowances issued by a competent authority of another Member State are recognised for the purpose of meeting an operator's, an aircraft operator's or a shipping company's obligations under paragraph 3.

Table 4: False Negative Error Case: High Lexical Overlap but Scope Expansion

fail, whereas generative LLMs are more likely to identify the semantic modification.

C Dataset Structure

Each entry in our dataset consists of a JSON object, which pairs an ALU from the old consolidated version with an ALU from the subsequent newer consolidated version. The schema for each entry defines the fields:

old_alu_id The identifier for the ALU in the old version. Art is short for Article, Par stands for Paragraph and Poi is abbreviated for Point.

new_alu_id The identifier for the ALU in the new version.

old_version_id The date at which the old consolidated version is dated to in YYYY-MM-DD format.

new_version_id The date at which the new consolidated version is dated to in YYYY-MM-DD format.

old_content The text corresponding to the old_alu_id in the old consolidated version.

new_content The text corresponding to the new_alu_id in the new consolidated version.

celex_id The celex identifier for the base legal act of the consolidated versions.

classification The change classification between as annotated according to the scheme in Appendix A.

Listing 1 shows an example dataset entry:

```
{
  "old_alu_id": "Art6_Par1",
  "new_alu_id": "Art6_Par1",
  "classification": "semantic_change",
  "old_content": "Member States shall take
      the necessary measures to ensure that
      all end-of life vehicles are stored
      (even temporarily) and treated in
      accordance with the general
      requirements laid down in Article 4 of
      Directive 75/442/EEC, and in compliance
      with the minimum technical requirements
      set out in Annex I to this Directive,
      without prejudice to national
      regulations on health and environment.",
  "old_version_id": "2017-12-06",
  "new_content": "Member States shall take
      the necessary measures to ensure that
      all end-of-life vehicles are stored
      (even temporarily) and treated in
      accordance with the waste hierarchy and
      the general requirements laid down in
      Article 4 of Directive 2008/98/EC of
      the European Parliament and of the
      Council, and in compliance with the
      minimum technical requirements set out
      in Annex I to this Directive, without
      prejudice to national regulations on
      health and environment.",
  "new_version_id": "2018-07-04",
  "celex_id": "32000L0053"
}
```

Listing 1: Example Dataset Entry

D Initial Alignment and Classification

For each pair of consecutive consolidated versions (V_{old}, V_{new}) of a regulation, we parse both documents into ALUs using a hierarchical legal structure (article - paragraph - point). Each ALU receives a stable structural identifier (e.g., `ArtX_ParY_PoiZ`), cleaned text, and metadata.

ALU alignment is performed in three passes:

1. **ID-based alignment:** ALUs with identical IDs across versions are paired directly.
2. **Similarity fallback:** unmatched ALUs in V_{new} are matched to unmatched ALUs in V_{old} using text similarity (Python Software Foundation, 2026), with a minimum threshold of 0.85.
3. **Residual handling:** remaining unmatched ALUs are labeled as `addition` (new-only) or `deletion` (old-only).

For aligned pairs, an initial rule-based change classification label is assigned:

- `no_change`: exact text match;

- `reference_update`: citation/footnote reference changes via regex expression;
- `semantic_change`: changes in numerical values, dates, or domain-critical units (e.g., g/km);
- `syntactic_change`: high textual similarity (greater than 0.85) without semantic triggers;
- otherwise: `semantic_change`.

To detect reference updates, we applied the case-insensitive pattern `(ISO|IEC|EN|EC|EU|Regulation|Directive|Decision)\s+(\d+[/-]\d+)` to both old and new ALU texts. The first capture group matches the reference type (e.g. `Directive`) and the second group matches the identifier format (e.g., `2018/858`, `123-45`). We extract all matches from each version, convert them to sets, and compare them. If the old and new reference sets differ and both are non-empty, the change is labeled `reference_update` rather than `syntactic_change` or `semantic_change`.

Importantly, this entire automated pipeline functioned exclusively as a pre-annotation step to scaffold the dataset creation. Every structural alignment and initial classification label produced here was subsequently subjected to manual verification by the human annotators.

E Dominance hierarchy of change

The dominance hierarchy is used to resolve competing change signals when multiple candidate labels are applicable. It formalizes the principle that labels with stronger legal effect dominate those with weaker effect.

The order of dominance is thus addition/deletion > semantic change > reference update > syntactic change > no change. For example, if an ALU change contains both a reference update and a semantic change, the resulting label would be a semantic change. This hierarchy also introduces the concept of similarity between the different change labels. A reference update label is "closer" to a syntactic change label than a semantic change label. If we consider additions and deletions as dominating other labels, due to the alignment process, we can also interpret the hierarchy as a scale from strongest change in legal meaning to weakest change.

F Multilingual Label Inference Rules

We infer non-English labels from English labels using a deterministic rule set motivated by legal equivalence across EU language versions. The goal is to preserve substantive change types while adjusting surface-level labels when text differs from English expectations.

Input For each aligned ALU pair, we use: (i) the English label, and (ii) the German old/new text pair.

Text identity check We treat German old/new text as *identical* if they match exactly after whitespace normalization.

Rules

- **Direct transfer rules.**
  - English `addition` → German `addition`
  - English `deletion` → German `deletion`
  - English `semantic_change` → German `semantic_change`
  - English `reference_update` → German `reference_update`
- **No-change adaptation**
  - English `no_change` + identical German text → `no_change`
  - English `no_change` + different German text → `syntactic_change`
- **Syntactic-change adaptation**
  - English `syntactic_change` + identical German text → `no_change`
  - English `syntactic_change` + different German text → `syntactic_change`

Fallback If none of the above cases applies, e.g. there is no matching ALU pair due to structural differences in the manifestations, we discard that pair.

G LLM Prompting Protocol and Templates

We defined prompt templates for all three tasks in our LLM baselines: (i) combined alignment+classification, (ii) alignment-only, and (iii) classification-only.

G.1 Output constraints

All templates enforce strict outputs:

- Combined: exactly two lines (`MATCH: ...`, `CLASSIFICATION: ...`)
- Alignment-only: exactly one line (`MATCH: ...`)
- Classification-only: exactly one line (`CLASSIFICATION: ...`)

No additional explanation text is requested.

G.2 Combined task templates

This section details the system prompts used when the LLM is instructed to perform alignment and classification simultaneously. Listing 2 presents the template for the forward direction (matching an old provision to a new one), while Listing 3 shows the reverse direction. Additionally, Listing 4 provides the dynamic user prompt template used for both directions.

```
You are an expert legal text analyst. Your
    task is to:
1. Find the best matching NEW legal
    provision segment from a list of
    candidate NEW legal provision segments
    for a given OLD legal provision segment
2. Classify the type of change between the
    matched legal provision segments

IMPORTANT: You are matching an OLD legal
    provision segment to NEW legal
    provision segments. If no good match is
    found, this means the OLD legal
    provision segment was DELETED.

Classify each change into exactly one of
    these categories:
1. "deletion" - The old legal provision
    segment was removed (no matching new
    segment found) - USE THIS IF NO MATCH
2. "no_change" - Text is identical or
    nearly identical
3. "semantic_change" - Substantive content
    change that alters the meaning
4. "syntactic_change" - Minor surface-level
    change that does not alter meaning
5. "reference_update" - Only legal
    references changed while core meaning
    is preserved

DO NOT use "addition" - that is only for
    reverse alignment (new -> old).

Respond in the following format:
MATCH: <index_of_best_match> or NONE
CLASSIFICATION: <classification_label>

If no good match exists, respond with:
MATCH: NONE
CLASSIFICATION: deletion

Output format rules (strict):
- Return EXACTLY two lines
- Line 1 must start with "MATCH:"
- Line 2 must start with "CLASSIFICATION:"
- Do not add explanations, bullets,
    markdown, or extra text
```

Listing 2: System prompt (forward: old -> new)

```
You are an expert legal text analyst. Your
    task is to:
1. Find the best matching OLD legal
    provision segment from a list of
    candidate OLD legal provision segments
    for a given NEW legal provision segment
2. Classify the type of change between the
    matched legal provision segments

IMPORTANT: You are matching a NEW legal
    provision segment to OLD legal
    provision segments. If no good match is
    found, this means the NEW legal
    provision segment was ADDED.

Classify each change into exactly one of
    these categories:
1. "addition" - The new legal provision
    segment was added (no matching old
    segment found) - USE THIS IF NO MATCH
2. "no_change" - Text is identical or
    nearly identical
3. "semantic_change" - Substantive content
    change that alters the meaning
4. "syntactic_change" - Minor surface-level
    change that does not alter meaning
5. "reference_update" - Only legal
    references changed while core meaning
    is preserved

DO NOT use "deletion" - that is only for
    forward alignment (old -> new).

Respond in the following format:
MATCH: <index_of_best_match> or NONE
CLASSIFICATION: <classification_label>

If no good match exists, respond with:
MATCH: NONE
CLASSIFICATION: addition

Output format rules (strict):
- Return EXACTLY two lines
- Line 1 must start with "MATCH:"
- Line 2 must start with "CLASSIFICATION:"
- Do not add explanations, bullets,
    markdown, or extra text
```

Listing 3: System prompt (reverse: new -> old)

```
Find the best matching {new|old} legal
    provision segment for this {old|new}
    legal provision segment and classify
    the change.

{OLD|NEW} LEGAL PROVISION SEGMENT:
"{query_content}"

CANDIDATE {NEW|OLD} LEGAL PROVISION
    SEGMENTS:
0. {candidate_0}
1. {candidate_1}
...

Respond with:
MATCH: <index> or NONE
CLASSIFICATION: <label>
```

Listing 4: User prompt template (combined)

G.3 Alignment-only templates

This section outlines the prompt templates used when evaluating the alignment-only task. Listing 5 shows the system prompt for the forward direction, Listing 6 provides the reverse direction, and Listing 7 displays the corresponding dynamic user prompt.

```
You are an expert legal text analyst. Your
    task is to find the best matching NEW
    legal provision segment from a list of
    candidate NEW legal provision segments
    for a given OLD legal provision segment.

Focus ONLY on finding the correct match. Do
    NOT classify the change type.

If the OLD legal provision segment was
    removed (no good match exists among the
    candidates), respond with NONE.

Respond in the following format ONLY:
MATCH: <index_of_best_match> or NONE

Output format rules (strict):
- Return EXACTLY one line
- The line must start with "MATCH:"
- Do not add explanations, bullets,
    markdown, or extra text
```

Listing 5: System prompt (alignment-only: forward)

```
You are an expert legal text analyst. Your
    task is to find the best matching OLD
    legal provision segment from a list of
    candidate OLD legal provision segments
    for a given NEW legal provision segment.

Focus ONLY on finding the correct match. Do
    NOT classify the change type.

If the NEW legal provision segment was
    added (no good match exists among the
    candidates), respond with NONE.

Respond in the following format ONLY:
MATCH: <index_of_best_match> or NONE

Output format rules (strict):
- Return EXACTLY one line
- The line must start with "MATCH:"
- Do not add explanations, bullets,
    markdown, or extra text
```

Listing 6: System prompt (alignment-only: reverse)

```
Find the best matching {new|old} legal
    provision segment for this {old|new}
    legal provision segment.

{OLD|NEW} LEGAL PROVISION SEGMENT:
"{query_content}"

CANDIDATE {NEW|OLD} LEGAL PROVISION
    SEGMENTS:
0. {candidate_0}
1. {candidate_1}
...

Respond with:
MATCH: <index> or NONE
```

Listing 7: User prompt template (alignment-only)

G.4 Classification-only templates

This section details the prompt templates used for the isolated classification task. Listing 8 provides the system prompt, and Listing 9 shows the user prompt template containing the old and new text versions.

```
You are an expert legal text analyst. You
    are given an OLD and NEW version of a
    legal provision segment. Classify the
    type of change between them.

Classify into exactly one of these
    categories:
1. "no_change"
2. "semantic_change"
3. "syntactic_change"
4. "reference_update"

Respond in the following format ONLY:
CLASSIFICATION: <classification_label>

Output format rules (strict):
- Return EXACTLY one line
- The line must start with "CLASSIFICATION:"
- Do not add explanations, bullets,
    markdown, or extra text
```

Listing 8: System prompt (classification-only)

```
Classify the type of change between these
    two versions of a legal provision.

OLD VERSION:
"{old_content}"

NEW VERSION:
"{new_content}"

Respond with:
CLASSIFICATION: <label>
```

Listing 9: User prompt template (classification-only)

Decoding and parsing. All LLM calls used deterministic decoding with temperature = 0 and strict response parsing for `MATCH` and `CLASSIFICATION` fields.

H Threshold Tuning and Train–Test Split

All experiments used a regulation-level split to avoid test set leakage:

- Training regulation: 32003L0087 (12 version-pair files, 3,973 entries)
- Test regulations: 32000L0053, 32007R0715, 32009R0595, 32019R0631, 32019R1242 (6 version-pair files, 799 entries)

Tuning objective For threshold-based baselines (all non-LLM baselines), thresholds were selected on the training split by maximizing macro-F1.

Combined-task tuning Grid search over:

- `threshold_high`
- `threshold_syntactic`
- `threshold_low`
- `alignment_threshold`

with constraint `threshold_low < threshold_syntactic < threshold_high`.

Classification-only tuning Grid search over:

- `threshold_high`
- `threshold_syntactic`
- `threshold_low`

again maximizing macro-F1 on train.

Search ranges and step sizes. We tuned thresholds with a simple grid search with values where `low < syntactic < high`.

- **Cosine** and **Sparse** (`bge-m3-sparse`, `splade`)

  Combined / alignment tasks

  - `high`: start 0.85, stop 1.00, step 0.01
  - `syntactic`: start 0.70, stop 0.98, step 0.02
  - `low`: start 0.40, stop 0.90, step 0.05
  - `alignment`: start 0.40, stop 0.95, step 0.05

  Classification-only task

  - `high`: start 0.85, stop 1.00, step 0.01

- `syntactic`: start 0.70, stop 0.98, step 0.02
- `low`: start 0.40, stop 0.90, step 0.05

- **Lexical** and **DocuToads-inspired**

  Combined / alignment tasks

  - `high`: start 0.80, stop 1.00, step 0.02
  - `syntactic`: start 0.45, stop 0.95, step 0.05
  - `low`: start 0.15, stop 0.75, step 0.05
  - `alignment`: start 0.20, stop 0.95, step 0.05

  Classification-only task

  - `high`: start 0.80, stop 1.00, step 0.02
  - `syntactic`: start 0.45, stop 0.95, step 0.05
  - `low`: start 0.15, stop 0.75, step 0.05

- **Sparse** (`bm25`, focused ranges)

  Combined / alignment tasks

  - `high`: start 0.80, stop 1.01, step 0.05
  - `syntactic`: start 0.20, stop 0.81, step 0.05
  - `low`: start 0.00, stop 0.31, step 0.05
  - `alignment`: start 0.00, stop 0.21, step 0.01

  Classification-only task

  - `high`: start 0.95, stop 1.01, step 0.005
  - `syntactic`: start 0.90, stop 1.001, step 0.005
  - `low`: start 0.00, stop 0.31, step 0.05

I Additional Results

I.1 Per class scores

This section contains more detailed per-class metrics on the alignment, classification, and combined tasks. Table 5 presents the test set per-class F1 scores for the combined task. Table 6 details the per-class F1 scores for the isolated classification task, and Table 7 reports the per-class accuracy for the structural alignment task. All results depict the performance of the threshold-tuned models.

Model Name	Add.	Del.	No Chg.	Ref.	Sem.	Syn.
Lexical & Sparse						
Levenshtein Ratio	0.9911	0.7310	0.9711	0.3333	0.1235	0.0000
DocuToads	0.9697	0.8068	0.9885	0.1176	0.4037	0.0000
BM25	1.0000	0.2054	0.0000	0.0000	0.0000	0.0000
splade	0.9911	0.7500	0.9671	0.0000	0.1446	0.0000
Dense Encoders						
bert-base-uncased	0.6122	0.6772	0.9458	0.4000	0.2136	0.0000
nlpaueb/bert-base-uncased-eurlex	1.0000	0.6261	1.0000	0.0000	0.0000	0.0000
all-MiniLM-L6-v2	0.9881	0.7978	0.9843	0.1818	0.4615	0.1667
all-MiniLM-L12-v2	0.9881	0.7912	0.9854	0.1538	0.3299	0.1429
paraphrase-mpnet-base-v2	0.9790	0.8092	0.9691	0.2500	0.2800	0.1667
paraphrase-multilingual-mpnet-base-v2	0.9728	0.8023	0.9843	0.1818	0.3929	0.2667
Qwen3-Embedding-0.6B	0.9697	0.7977	0.9593	0.3333	0.2105	0.0000
Qwen3-Embedding-4B	0.9634	0.8092	0.9711	0.0000	0.3000	0.0000
Generative LLMs						
Llama3-8b-instruct	0.8372	0.2025	0.0408	0.0000	0.1096	0.0000
DeepSeek-V3.2	0.9068	0.7785	0.9620	0.3636	0.3939	0.0000
Llama-4-Maverick-17B-128E-Instruct	0.7943	0.5976	0.7847	0.0000	0.0625	0.0307
gpt-oss-120b	0.8997	0.6145	0.8899	0.2308	0.5035	0.0526

Table 5: Test set per class F1 scores for combined task

Model Name	No Chg.	Ref.	Sem.	Syn.
Lexical & Sparse				
Levenshtein Ratio	0.9916	0.1111	0.8120	0.3077
DocuToads	0.9958	0.1111	0.8209	0.2500
BM25	0.0000	0.1333	0.2126	0.0000
splade	0.9916	0.1333	0.7813	0.2857
Dense Encoders				
bert-base-uncased	0.9802	0.2857	0.7460	0.1000
nlpaueb/bert-base-uncased-eurlex	1.0000	0.1333	0.8800	0.0000
all-MiniLM-L6-v2	0.9874	0.1538	0.7059	0.1429
all-MiniLM-L12-v2	0.9854	0.1429	0.7227	0.2400
paraphrase-mpnet-base-v2	0.9854	0.2000	0.8088	0.0000
paraphrase-multilingual-mpnet-base-v2	0.9843	0.1818	0.8030	0.2857
Qwen3-Embedding-0.6B	0.9554	0.2857	0.6182	0.0000
Qwen3-Embedding-4B	0.9864	0.2222	0.7937	0.0833
Generative LLMs				
Llama3-8b-instruct	0.9925	0.2857	0.7541	0.2564
DeepSeek-V3.2	0.9989	0.6000	0.9730	0.9231
Llama-4-Maverick-17B-128E-Instruct	0.9925	0.5714	0.9801	0.6667
gpt-oss-120b	0.9979	0.6000	0.8986	0.9333

Table 6: Test set per-class F1 scores for the classification-only task (excluding addition/deletion)

Model Name	Add.	Del.	No Chg.	Ref.	Sem.	Syn.
Lexical & Sparse						
Levenshtein Ratio	0.9765	0.7500	0.8726	0.5000	0.3867	0.7143
DocuToads	0.9353	0.7222	0.9002	0.7500	0.5200	0.8571
BM25	1.0000	1.0000	0.0000	0.0000	0.0000	0.0000
splade	0.9647	0.7500	0.8854	0.7500	0.4667	0.8571
Dense Encoders						
bert-base-uncased	0.4471	0.4306	0.9130	0.7500	0.5200	0.5714
nlpaueb/bert-base-uncased-eurlex	0.9941	0.7639	0.8323	0.2500	0.1200	0.0000
all-MiniLM-L6-v2	0.9706	0.7222	0.9023	0.7500	0.5733	0.8571
all-MiniLM-L12-v2	0.9706	0.7361	0.8981	0.7500	0.5200	0.8571
paraphrase-mpnet-base-v2	0.9529	0.7083	0.9108	0.7500	0.6000	0.8571
paraphrase-multilingual-mpnet-base-v2	0.9412	0.6944	0.9108	0.7500	0.6000	0.8571
Qwen3-Embedding-0.6B	0.9353	0.6667	0.9193	0.7500	0.5600	0.8571
Qwen3-Embedding-4B	0.9176	0.6806	0.9193	0.7500	0.6000	0.8571
Generative LLMs						
Llama3-8b-instruct	0.2800	0.1562	0.5486	0.0000	0.6250	0.2500
DeepSeek-V3.2	0.8176	0.3889	0.8599	0.7500	0.7200	0.7143
Llama-4-Maverick-17B-128E-Instruct	0.3706	0.0972	0.8450	0.5000	0.6000	0.5714
gpt-oss-120b	0.7000	0.3750	0.8153	0.5000	0.6533	0.7143

Table 7: Test set per-class alignment accuracy