# Fine-Grained Semantic Comparison of Legal Documents using LLMs

**Elisei Rykov[1,*], Nikolay Ivanov[1,*], Maria Bandulevich[2], Kseniia Petrushina[1], Valentin Malykh[3,4], Vasily Konovalov[5,6], Alexander Panchenko[1,5], Ilseyar Alimova[1]**

[1]Applied AI Institute, [2]HSE, [3]MWS AI,
[4]Trusted AI Research Center, RAS, [5]AXXX, [6]MIRAI

**Correspondence:** nikolayivanov1999@gmail.com

## Abstract

Frequent revisions of complex regulatory documents in large organizations often introduce inconsistencies and contradictions that are difficult for lawyers and auditors to detect manually. Existing tools rely on character-level diffs and therefore miss paraphrases and semantic shifts. We introduce LegDiff, a novel benchmark for evaluating span-aware semantic comparison of legal texts, and use it to investigate the ability of large language models to detect semantic changes beyond token- and character-level matching. LegDiff comprises manually annotated pairs of legal paragraphs drawn from different documents. In addition, we present a pipeline to generate synthetic training data that aligns with the manual annotations and mirrors the structure and label distribution of the manually curated benchmark, and a visualization tool for clearly displaying detected differences and inconsistencies. The dataset and code are publicly available to facilitate reproducibility and further research.[1]

**Document A:**

For an income payer (an organization or a sole proprietor) paying a foreign citizen or a stateless person registered under Article 83(7.4), the procedure for sending a notice confirming the foreign citizen's tax registration also applies when the relevant information is received via electronic channels, and additionally requires the tax authority to provide the specified notice in hard copy upon request.

**Document B:**

For an income payer (an organization or a sole proprietor) paying a foreign citizen or a stateless person registered under Article 83(7.4), the procedure for sending an extract from the Unified State Register of Taxpayers containing the foreign citizen's tax registration information also applies when the relevant information is received via electronic channels.

Figure 1: Example of a revision in a legal document from the proposed LegDiff dataset. Colors indicate span-level semantic relation classes: green spans are **Equivalent** paraphrases, red spans are **Contradiction** between a tax-registration notice and a taxpayer-register extract, and the blue span is an **Addition** requiring hard-copy provision upon request.

## 1 Introduction

Organizations rely on standard templates for contracts, administrative documents and regulatory materials such as policies, standards and guidelines that define processes and structure and are adapted for specific projects. These documents are used by employees to perform tasks and by auditors to verify compliance, but in large companies frequent revisions often create conflicts, inconsistencies or contradictions between rules or versions and thereby increase the risk of errors, delays, asset compromise, regulatory violations and fraud. When preparing a final version, it is therefore important not merely to replace fields and particulars but to compare the final text with the approved template, identify deviations, check for internal inconsistencies and ensure that no new material terms related to finance, timing, risk or warranties have been added that could affect the legal status of the transaction. Any detected changes must be reviewed and agreed by the responsible departments before signing in order to mitigate legal and commercial risks. Manual detection of such issues is particularly challenging because these documents are often lengthy, written in dense legal language and exist in multiple frequently updated versions, which increases the likelihood of overlooked changes.

The simplest approaches to solve this problem involve a straightforward character-by-character comparison of two documents to identify differ-

[*]These authors contributed equally to this work.
[1]https://github.com/s-nlp/SLeDoC



*Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 982–994
July 2-7, 2026 

ences. The examples of such tools are CopyLeaks[2], Draftable[3], Plagiarism Checker X[4], Aspose AI[5], Embedika[6], iThenticate[7], and Sidekicker[8]. However these tools are limiting their use to highly overlapping texts and failing with paraphrases or dispersed similarities. Other approaches perform cross-document inference (e.g., (Yuan et al., 2025)), extracting contradictory or entailed expressions, or identify contradictions between company and license documents, as in ContractNLI (Koreeda and Manning, 2021). However, these methods focus on specific relation types and lack comprehensive text analysis.

In this work, we address the problem of semantically comparing two legal documents at the paragraph level. We propose an approach that compares paragraphs from different document versions and identifies contradictions, additions, and equivalent fragments. Unlike prior methods, our approach analyzes smaller semantic fragments, or spans, rather than whole sentences, which is important in the legal domain where long and complex constructions often occur within a single sentence.

We introduce LegDiff, a manually annotated benchmark of Russian paragraph pairs drawn from legal codes, segmented into spans and labeled with the corresponding relation types: Contradiction, Addition and Equivalent.

The example of two paragraphs from LegDiff corpus is presented in Figure 1. We investigate methods leverages modern large language models, enabling cross-lingual and cross-domain application and allowing it to recognize semantic correspondences in challenging cases. Finally, we present a procedure for generating synthetic training data based on naturally occurring legislative revisions.

Thus, the contributions of this paper are as follows: (i) LegDiff, a novel benchmark for span-level comparison of legal texts; (ii) a pipeline for synthetically generating training data aligned with manual annotations; (iii) extensive experiments with large language models of various architectures and operating modes on the LegDiff dataset.

[2] https://app.copyleaks.com
[3] https://draftable.com
[4] https://justdone.com/plagiarism-checker
[5] https://products.aspose.ai/total/ai-document-comparison
[6] https://compare.embedika.ru
[7] https://www.ithenticate.com
[8] https://sidekicker.ai/plagiarism-checker

## 2 Related Work

In the legal domain, most research on document comparison focuses on detecting contradictions. ContractNLI (Koreeda and Manning, 2021) adapts natural language inference to contracts to capture contradictions between contractual obligations and external license agreements, and Schumann et al. (Schumann et al., 2026) show that LLMs can achieve promising results in detecting contradictions and inconsistencies in regulatory texts. LegalLens addresses two core tasks: detecting legal violations in unstructured text and linking those violations to potentially affected individuals (Bernsohn et al., 2024).

Reviews of methods for finding redundancy, circularity, or inconsistency indicate that the majority of approaches operate at the sentence level (Schumann and Gómex, 2024). These findings support our choice to treat legal revisions at the span level rather than only at the sentence or paragraph level. Restricting context to single sentences often causes cross-sentence coreferences to be overlooked and prevents checking whether the missing information behind an inconsistency appears in adjacent sentences. Moreover, these studies generally emphasize algorithm development without providing visualizations, which limits their practical adoption by end users. Among visualization systems, LegalViz stands out by producing simple Graphviz diagrams that highlight entities, transactions, legal sources, and key statements in judgments (Resck et al., 2023), but it is geared toward precedent citation search and visualization rather than comprehensive semantic alignment.

Beyond the legal domain, related work has examined semantic comparison and alignment across texts. Comparative Document Analysis aims to surface commonalities and distinctions via phrase-level comparison, while semantic matching methods learn representations for comparing texts (Ren et al., 2017; Jiang et al., 2019). However, these approaches typically operate at a coarser level than the localized edits required for semantic diff analysis.

Research on factual inconsistency detection investigates contradictions and inconsistencies between text fragments, focusing on detection and mitigation, but it generally does not produce explicit span-to-span alignments with interpretable relation labels (Lattimer et al., 2023). Such methods primarily target coarse document similarity,

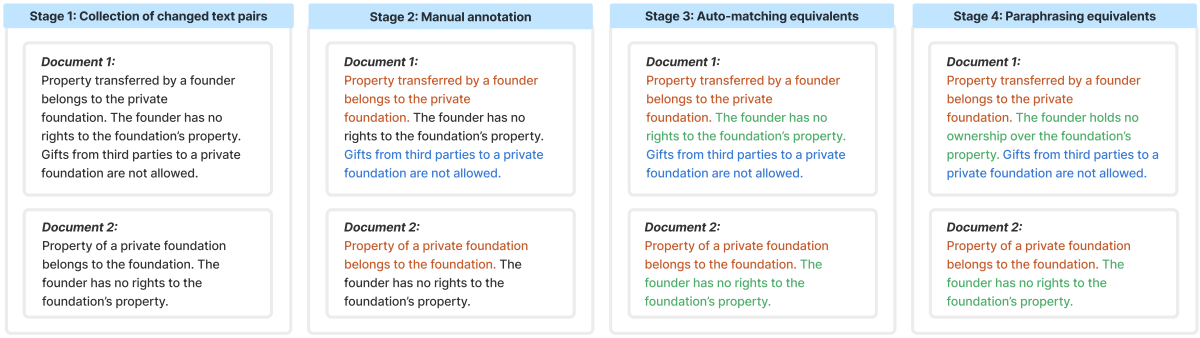


Figure 2: LegDiff construction pipeline: (i) collect temporally revised legal provisions, (ii) manually annotate modified spans as **Contradiction** or **Addition**, (iii) automatically align unchanged spans as **Equivalent**, and (iv) paraphrase **Equivalent** spans to reduce trivial lexical overlap.

phrase extraction, or inconsistency spotting rather than end-to-end alignment of semantically corresponding spans.

Another closely related line of work focuses on fine-grained semantic decomposition. PropSegmEnt (Chen et al., 2023) introduces proposition-level segmentation and semantic relation classification on English news and Wikipedia text, demonstrating the value of breaking complex text into smaller semantic units. However, it treats segmentation and classification as separate tasks, uses non-contiguous proposition spans, and does not model the end-to-end span alignment setting studied here. Similarly, recent studies emphasize span-level reasoning and interactions between fragments as an important component of natural language understanding (Ray Choudhury et al., 2023). These findings motivate our choice to treat revisions at the span level. Previous studies on Russian legal documents primarily focus on rule-based information extraction, rather than on document comparison or cross-text alignment (Khasianov et al., 2018). Nevertheless, existing resources do not cover the full setting we address, which includes pairwise span-to-span alignment with multi-class semantic labels. Thus, our work integrates these components into a unified system for semantic diff analysis of legal texts.

## 3 LegDiff Dataset

In this section, we present a novel manually annotated benchmark for span-level semantic matching in the legal domain LegDiff, and a methodology for synthetic training data generation for the task of semantic matching.

### 3.1 LegDiff Construction Pipeline

We construct LegDiff following the four stages illustrated in Figure 2. First, we collected temporally changed laws from Russian Codes, with changes introducing new constraints or altering information. These changes are natural editions of different legal codes. The source texts were obtained from the Consultant+ website[9]. For identifying changes between document versions we used the service's built-in comparison editor, which performs a straightforward character-level comparison of texts. The editor's results were used to mark the modified paragraphs. Second, each pair of changed paragraphs was manually annotated by extracting the modified spans and labeling it as Contradiction or Addition following Guidlines provided in section 3.2.

Two annotators proficient in Russian annotated the data. We report inter-annotator agreement as raw percent agreement before adjudication, which was 71.2%. Any disagreements were then resolved and the final annotation was confirmed by a moderator with a background in natural language processing. As a result of this step, for each paragraph we produced lists of span pairs labeled as Contradiction and Addition. At the third stage we automatically extracted all spans left unannotated in the prior step.

For every unannotated span in one document, the procedure searched for a matching span in the other document and assigned it the label Equivalent. Finally, we paraphrase spans of Equivalent using Claude Opus-4.6 to reduce reliance on surface string overlap and encourage semantic matching. The final paraphrased spans underwent manual review and validation, and any detected inaccuracies

[9] https://www.consultant.ru

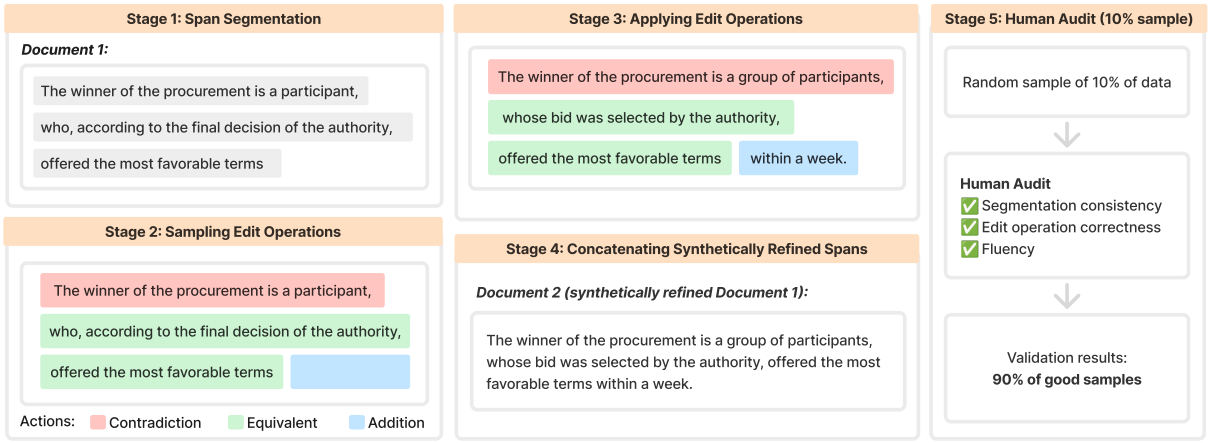


Figure 3: LegDiff-Synth construction pipeline: (i) we collect paragraphs from the legal documents, (ii) segment them into spans and associated atomic claims, sample edit operations per span (paraphrase for Equivalent and edits for Contradiction), (iii) sample Addition placeholders between spans, and (iv) apply the sampled operations to obtain a revised document.

were fixed. We release the paraphrasing prompt in the repository for reproducibility.

### 3.2 Annotation Guidelines

The annotation procedure aims to identify spans in two paragraphs that express semantic equivalence, contradiction, or addition. Annotators mark minimal spans that capture the relevant semantic relation between the fragments. In this section, we provide a shortened version of the annotation guidelines, the full version is available in the project repository.

Equivalent spans include all semantically matching parts of the two paragraphs. These may contain: identical fragments and paraphrased fragments conveying the same meaning. All elements participating in the shared meaning should be included in the span, even if they differ lexically but preserve the same semantics.

Contradiction spans include fragments that refer to the same semantic position in a statement (e.g., object of an action, condition, basis, alternative) but contain incompatible or mutually exclusive values. For contradictions expressed at the level of short phrases or expressions, annotators mark the minimal span containing only the differing parts. Identical words that do not affect the interpretation of the contradiction are excluded. If the contradiction is expressed in a larger syntactic unit (e.g., a clause with a grammatical base, a one-member sentence, or a fixed expression), the entire fragment containing the contradiction is marked. In some cases, lexically identical elements remain inside the span if they determine the structure of the statement (e.g., specifying the recipients or participants of an action). Such elements are included even if they appear in both paragraphs.

| Span type | # Span pairs | Avg. length (chars) |
|---|---|---|
| Equivalent | 262 | 196.8 |
| Contradiction | 85 | 71.9 |
| Addition | 111 | 112.1 |
| All | 458 | 153.1 |

Table 1: Overall statistics of categories annotated in the proposed LegDiff dataset.

Addition spans include fragments from one paragraph that introduce new information relative to a matching or equivalent fragment in the other paragraph. The added content does not contradict the original meaning but specifies, expands, clarifies, or introduces an additional condition, limitation, or consequence. Additions may appear as a sentence component, a predicate complement, a separate grammatical construction, a new sentence, or a combination of these forms.

### 3.3 Dataset Statistics

Overall statistics of the LegDiff dataset by span type are presented in Table 1. In total, LegDiff contains 154 paragraph pairs in Russian, drawn from nine Russian legal codes: Forest, Arbitration Procedure, Civil, Labour, Housing, Tax, Maritime, Budget, and Water, yielding 458 annotated span pairs in total.

The distributions of span pairs by type and paragraph pairs by legal code are shown in Figure 4. Equivalent spans are the longest on average

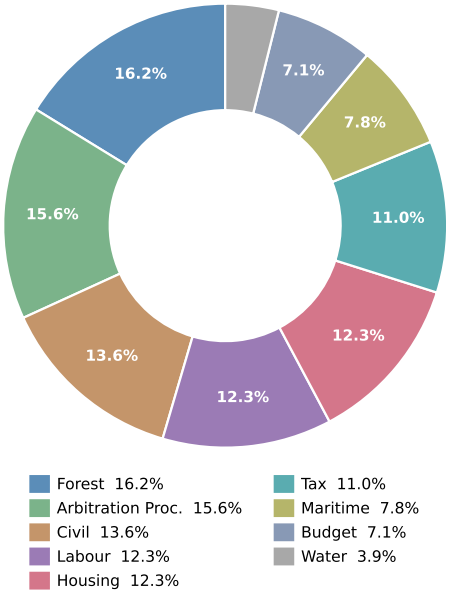


Figure 4: Distribution of paragraph pairs by legal code in the LegDiff dataset.

(196.8 characters), while Contradiction spans are the shortest (71.9 characters), consistent with the observation that contradictions are typically localized to short differing phrases.

### 3.4 LegDiff Synthetic

Since LegDiff is a manually annotated benchmark and manual labeling is time-consuming and expensive, we introduce a pipeline for synthetic data collection in a similar format, using only parsed Russian laws from the Criminal Code, the Customs Code, and the Family Code. We use different codes than LegDiff to avoid overlap. The overall pipeline of dataset constraction is shown in Figure 3. In this paper, we refer to the dataset as LegDiff-Synth.

The pipeline consists of five stages. First, paragraphs from different legal codes are collected. Second, we perform two-level segmentation: (i) split the text into contiguous spans, and (ii) generate an atomic claim for each span. This provides a claim-level representation while preserving the original span boundaries. Using claim-level representations encourages minimal, localized edits, which is crucial for producing realistic revisions where most of the text remains unchanged. Third, we sample an edit operation for each span: Equivalent (implemented via paraphrasing) or Contradiction (implemented via targeted edits). Independently, we sample Addition placeholders between spans. Fourth, we apply the sampled operations to generate modified spans. Fifth, we concatenate the

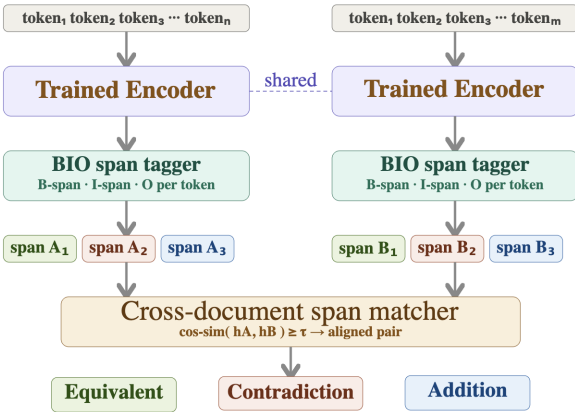


Figure 5: Encoder-based approach overview.

modified spans into a revised paragraph.

For the LegDiff-Synth dataset, we constructed 907 paragraph pairs with controlled synthetic revisions in Russian. At generation time, each source span is assigned either Equivalent or Contradiction, while Addition spans are sampled independently in the gaps between neighboring spans. Concretely, we sample Contradiction per span with probability 25% and insert Addition placeholders per inter-span gap with probability 20%, so the resulting corpus is dominated by Equivalent spans, with Addition and Contradiction appearing as less frequent edited cases in proportions chosen to match the manual LegDiff benchmark.

## 4 Experiments

We evaluated the methods on the LegDiff corpus using an end-to-end span-level metric $F1_{\text{SLM}}$ that penalizes both span-matching and label errors. Predicted spans are aligned to gold spans via a similarity function with threshold, after which per-class $F_1$ is computed over the aligned pairs; we report per-class $F1_{\text{SLM}}$. We use token-set Jaccard as sim with $\tau = 0.8$.

### 4.1 Models

We evaluate three methods: a difflib baseline, an LLM-based approach, and an encoder-based approach. First, `difflib` is a string-level baseline using Python's `difflib.SequenceMatcher`. Its edit ops are mapped to our labels as follows: equal - Equivalent, replace - Contradiction, insert/delete - Addition.

Second, LLMs were evaluated in two setups: few-shot prompting (FS; the prompt directly asks the model to segment each paragraph pair into spans, align them, and label each aligned pair as

Contradiction, Addition or Equivalent without task-specific weight updates) and LoRA fine-tuning on LegDiff-Synth dataset (FT; rank 64, 4 epochs) (Hu et al., 2022). The prompt was identical for all models; the full prompt is available in the Appendix B. We employed both models with publicly available weights in different sizes (LLaMA3.1-8B[10], Qwen2.5-Instruct-7B[11], Qwen2.5-32B-Instruct[12]) and closed, proprietary models such as Claude and ChatGPT.

Third, the encoder-based model was fine-tuned for span tagging and matching (4 epochs) as presented in Figure 5. In this method, both documents are passed through a shared-weight encoder, a BIO span tagger then extracts contiguous spans from each document's token representations. The extracted spans are mean-pooled into fixed-size embeddings and aligned cross-document via cosine similarity with threshold $\tau = 0.8$, after which each aligned pair is classified into one of the three relation labels using a linear layer. We chose mmBERT as the encoder, as it is currently the state-of-the-art among pre-trained encoder models (Marone et al., 2025).

### 4.2 Evaluation Metric

To evaluate methods on LegDiff, we define an end-to-end span-level metric, $F1_{\text{SLM}}$, which penalizes both span matching errors and label misclassification. The metric first aligns predicted span pairs with gold span pairs using a similarity function and threshold, and then computes standard per-class $F_1$ over the aligned pairs.

Let $D_1$ and $D_2$ be two documents. A system outputs a set of labeled aligned span pairs, and we denote gold and predicted annotations as

$$G(D_1, D_2) = \{(g_i^1, g_i^2, y_i)\}_{i=1}^{|G|}, \quad (1)$$

$$P(D_1, D_2) = \{(p_j^1, p_j^2, \hat{y}_j)\}_{j=1}^{|P|}, \quad (2)$$

where $g_i^1, p_j^1 \subset D_1$ and $g_i^2, p_j^2 \subset D_2$ are contiguous spans, and $y_i, \hat{y}_j \in L$ with $L = \{\texttt{equivalent}, \texttt{contradiction}, \texttt{addition}\}$.

**Span alignment.** We use a similarity function $\text{sim}(\cdot, \cdot) \in [0, 1]$ and a threshold $\tau \in (0, 1]$. A gold tuple $g_i$ and a prediction $p_j$ are considered aligned if both spans match:

$$\min\left(\text{sim}(g_i^1, p_j^1),\ \text{sim}(g_i^2, p_j^2)\right) \geq \tau \quad (3)$$

[10] meta-llama/Llama-3.1-8B-Instruct
[11] Qwen/Qwen2.5-7B-Instruct
[12] Qwen/Qwen2.5-32B-Instruct

| Method (params) | Mode | Contr. | Add. | Equiv. | Avg |
|---|---|---|---|---|---|
| difflib baseline (–) | – | 7.64 | 34.78 | 23.18 | 21.87 |
| Llama3.1 (8B) | FS | 5.06 | 30.86 | 41.36 | 25.76 |
| Qwen2.5-Instruct (7B) | FS | 3.01 | 28.93 | 34.78 | 22.24 |
| Qwen2.5-Instruct (32B) | FS | 10.58 | 45.00 | 46.35 | 33.98 |
| Llama3.1 (8B) | FT | 5.00 | 45.54 | 63.22 | 37.92 |
| Qwen2.5-Instruct (7B) | FT | 7.55 | 46.39 | 63.46 | 39.13 |
| Qwen2.5-Instruct (32B) | FT | 10.06 | 50.93 | 64.62 | 41.87 |
| mmBERT (0.37B) | FT | 3.60 | 27.91 | 33.82 | 21.78 |
| Claude Opus 4.6 (–) | FS | 23.23 | 70.00 | **53.33** | 48.85 |
| GPT-5.2 (–) | FS | **29.79** | **72.53** | 52.73 | **55.65** |

Table 2: Performance on LegDiff dataset on end-to-end semantic matching across three relation types: Contradiction (Contr.), Addition (Add.), and Equivalence (Equiv.). Results are shown for different models and training modes: few shot (FS) inference and fine-tuning (FT). Bold values indicate the best result within each relation, while underlined values denote the best performance among frozen models.

If multiple predictions satisfy (3) for a given $g_i$, we select the prediction with the highest matching score (ties are broken arbitrarily), and each prediction can be matched to at most one gold tuple.

**Counting TP/FP/FN.** After matching, an aligned pair $(g_i, p_j)$ is a true positive for class $c \in L$ if $y_i = c$ and $\hat{y}_j = c$. If $y_i \neq \hat{y}_j$, the gold contributes a false negative for its class and the prediction contributes a false positive for its predicted class. Unmatched gold tuples count as false negatives for their gold class, and unmatched predictions count as false positives for their predicted class. Finally, we compute per-class $F1_{\text{SLM}}$ using standard definitions from the resulting TP/FP/FN counts.

## 5 Results

The results on LEGDIFF are shown in Table 2. The experiments reveal a clear performance hierarchy and highlight three main patterns: (i) proprietary few-shot LLMs substantially outperform other approaches on semantic shifts, (ii) fine-tuning on synthetic data markedly improves open-weight models, and (iii) simple surface baselines and the encoder-based mmBERT struggle with deep semantic phenomena. Importantly, FS and FT probe different capabilities: FS measures what a model can do from instruction following alone, whereas FT measures how well an open-weight model can adapt when exposed to task-specific synthetic supervision. Below we provide a detailed analysis of the obtained results.

**Difflib baseline.** The `difflib` baseline achieves only Avg $F1_{SLM} = 21.87\%$, with poor performance across classes (Contradiction 7.64%, Addition 34.78%, Equivalent 23.18%). These results demonstrate that difflib cannot handle paraphrasing or semantic shifts, confirming the inadequacy of surface-level comparisons.

**Few-shot LLMs.** Among the few-shot models, GPT-5.2 achieves the best overall performance (Avg $F1_{SLM} = 55.65\%$), with the highest scores in Contradiction (29.79%) and Addition (72.53%). Claude Opus 4.6 is close behind (Avg 48.85%) and leads on Equivalent (53.33%), suggesting that it is more conservative in labeling spans as novel or conflicting. Open-source few-shot models lag considerably: Qwen2.5-Instruct (Team, 2024) (32B) reaches Avg 33.98%, while smaller models (Qwen2.5-7B, LLaMA3.1-8B) score around Avg 22%–26%, showing a clear gap relative to proprietary models in the zero-resource setting.

**Effect of fine-tuning on LegDiff-Synth.** Fine-tuning on LegDiff-Synth substantially improves all open-weight models. Qwen2.5-Instruct-32B improves from Avg 33.98% (FS) to 41.87% (FT), with gains in both Addition (45.00% → 50.93%) and Equivalent (46.35% → 64.62%). Smaller models benefit even more proportionally: Qwen2.5-7B jumps from 22.24% to 39.13% and LLaMA3.1-8B from 25.76% to 37.92%, underscoring that even modest models profit substantially from task-specific synthetic training data. Notably, the synthetic Equivalent paraphrases were generated with Claude Opus 4.6, yet Claude does not obtain the best overall score on LegDiff; GPT-5.2 remains stronger on average. This suggests that the data construction pipeline does not introduce a trivial bias toward the model used for paraphrase generation.

**Encoder-based approach.** The fine-tuned mmBERT-base model (Avg 21.78%) performs below all other fine-tuned models and is comparable to the `difflib` baseline, with a substantial decrease in $F1_{SLM}$ in all classes. Despite its small size (0.37B), its span-tagging objective struggles with the deep semantic understanding required for this task, particularly for Contradiction (3.60%).

**Visualization.** The proposed approach was integrated into a legal document comparison and visualization system described in (Rykov et al., 2026). The system provides an interactive interface for exploring span-level alignments and automatically generated explanations of detected relations between document fragments. This enables more interpretable analysis of complex legal revisions and facilitates qualitative inspection of model outputs.

## 6 Ablation Study

In addition to the main experiments, we evaluated our approach on the LegDiff dataset without paraphrasing equivalent fragments (i.e., before step 4 of the dataset construction pipeline). This setting allowed us to assess how well modern language models can identify identical text segments in the source texts. To further evaluate the robustness of the proposed approach, we also conducted experiments in a general-domain setting using the English PropSegment dataset (Chen et al., 2023). Additionally, we investigated the sensitivity of the results to the chosen span similarity threshold $\tau$.

| Method (params) | Mode | Contr. | Add. | Equiv. | Avg |
|---|---|---|---|---|---|
| difflib baseline (–) | – | 40.19 | 60.22 | **100.00** | 66.71 |
| Llama3.1 (8B) | FS | 5.48 | 33.52 | 48.97 | 29.32 |
| Qwen2.5-Instruct (7B) | FS | 4.58 | 29.09 | 43.26 | 25.64 |
| Qwen2.5-Instruct (32B) | FS | 23.12 | 46.53 | 61.21 | 43.62 |
| Llama3.1 (8B) | FT | 10.70 | 39.17 | 86.94 | 45.60 |
| Qwen2.5-Instruct (7B) | FT | 14.46 | 45.32 | 86.43 | 48.74 |
| Qwen2.5-Instruct (32B) | FT | 22.12 | 54.71 | 89.64 | 55.49 |
| mmBERT (0.37B) | FT | 7.77 | 31.78 | 35.66 | 25.07 |
| Claude Opus 4.6 (–) | FS | 40.76 | 68.20 | 96.28 | 68.41 |
| GPT-5.2 (–) | FS | **44.44** | **70.51** | 96.44 | **79.59** |

Table 3: Performance on the non-paraphrased version of the LegDiff dataset.

### 6.1 LegDiff non-paraphrased experiments

Table 3 reports results on the non-paraphrased version of LegDiff, where the equivalent spans retain their original surface form from the source documents. As expected, this setting is substantially easier for all models: the `difflib` baseline reaches an Avg of 66.71%, with a perfect Equivalent score (100.00%), since identical strings trivially match.

Among LLMs in the few-shot setting, Claude Opus 4.6 and GPT-5.2 both achieve strong performance, with GPT-5.2 attaining the best overall Avg (79.59%) and leading both in Contradiction (44.44%) and Addition (70.51%). Claude Opus 4.6 is close behind (Avg 68.41%), with competitive Contradiction (40.76%) and Addition (68.20%) scores.

Open source models improve considerably compared to the paraphrased setting: fine-tuned Qwen2.5-Instruct-32B reaches the highest Avg among fine-tuned models (55.49%), with strong Equivalent (89.64%) and Addition (54.71%) scores. Across settings, the gap between fine-tuned open-weight models and proprietary few-shot models narrows on non-paraphrased examples but does not disappear, suggesting that proprietary models retain an advantage on semantic reasoning beyond simple lexical matching. However, the encoder-based approach with mmBERT substantially degraded in this setting (Avg 25.07%), likely because its span-tagging objective works poorly in such deep context understanding.

| Method (params) | Mode | Contr. | Add. | Equiv. |
|---|---|---|---|---|
| T5-Large (0.7B) | FT | 20.34 | **93.98** | 84.78 |
| Qwen2.5-Instruct (32B) | FS | 9.00 | 88.00 | 79.00 |
| Qwen2.5-Instruct (32B) | FT | **38.30** | 93.51 | **85.44** |
| GPT-4o (-) | FS | 12.35 | 90.41 | 80.04 |
| mmBERT (0.37B) | FT | – | 89.34 | 74.18 |

Table 4: Results on PropSegmEnt dataset. FS stands for Few-Shot and FT for Fine-Tuning. T5-Large scores are from the original PropSegmEnt paper. The performance of mmBERT on Contradiction is not reported as PropSegmEnt contains too few Contradiction examples. In PropSegmEnt, the semantic matching task is divided into two: segmentation and classification. Therefore, the ground truth segments were used for evaluation.

### 6.2 PropSegment Experiments

PropSegmEnt comprises news articles and Wikipedia entries in English. The benchmark separates segmentation and semantic-relation classification into two distinct leaderboards. The goal of segmentation in PropSegmEnt is to sample diverse claims consisting of non-contiguous spans represented as a set of tokens from the input text. There are two main metrics for segmentation: the percentage of successfully matched spans using Jaccard similarity with a 0.8 threshold and the percentage of exactly matched spans. The prompt for the segmentation task is provided in C. The second task involves classifying the semantic relationship between the sampled claims and the original text. PropSegmEnt contains very few (near-zero) Contradiction instances, hence Contradiction scores are not statistically meaningful. In PropSegmEnt, semantic relation classification uses Entailment, Contradiction, and Neutral.

The PropSegmEnt results on semantic relation classification for Equivalent, Contradiction, Neutral classes are shown in Table 4. Our decoder-based and encoder-based approaches are shown in contrast to the original T5-Large method reported in Chen et al. (2023). We demonstrate that fine-tuned Qwen2.5-Instruct performs best on Equivalent and Contradiction, while T5-Large remains slightly stronger on Neutral. As with LegDiff, the encoder-based approach with the mmBERT-base model demonstrates worse results. The near-zero Contradiction scores are expected given the scarcity of contradiction instances in PropSegmEnt. The prompt for the entailment classification task is also provided in D. Results for segmentation task are presented in Appendix A.

| Method (params) | Mode | Contr. | Add. | Equiv. | Avg |
|---|---|---|---|---|---|
| difflib baseline (–) | – | 14.29 | 44.00 | 18.52 | 25.60 |
| Llama3.1 (8B) | FS | 5.88 | 0.00 | 23.81 | 9.89 |
| Qwen2.5-Instruct (7B) | FS | 0.00 | 18.18 | 18.60 | 36.78 |
| Qwen2.5-Instruct (32B) | FS | 8.45 | 33.90 | 40.00 | 27.45 |
| Claude Opus 4.6 (–) | FS | 27.27 | 55.17 | **55.70** | 46.04 |
| GPT-5.2 (–) | FS | **27.91** | **73.68** | 51.85 | **51.14** |

Table 5: Performance on the paraphrased version of the LegDiff-en dataset.

| Method (params) | Mode | Contr. | Add. | Equiv. | Avg |
|---|---|---|---|---|---|
| difflib baseline (–) | – | **43.59** | 59.34 | **97.83** | **66.92** |
| Llama3.1 (8B) | FS | 6.78 | 17.39 | 31.82 | 18.66 |
| Qwen2.5-Instruct (7B) | FS | 0.00 | 8.08 | 18.60 | 8.89 |
| Qwen2.5-Instruct (32B) | FS | 5.97 | 46.67 | 44.21 | 31.61 |
| Claude Opus 4.6 (–) | FS | 37.50 | 64.52 | 87.76 | 63.26 |
| GPT-5.2 (–) | FS | 41.18 | **65.22** | 88.24 | 64.88 |

Table 6: Performance on the non-paraphrased version of the LegDiff-en dataset.

### 6.3 LegDiff-en Results

To evaluate the robustness of the proposed method, we conducted experiments on the English-language LegDiff-en dataset, annotated using the same procedure as the Russian dataset. The annotation was performed by the same annotator, and the dataset consists of texts from English legal codes, ensuring consistency in the labeling scheme across languages. In total, the dataset contains 44 paragraph pairs, with 78 Equivalent spans, 40 Addition spans, and 31 Contradiction spans.

The results on LegDiff-en experiments presented in Table 5 and Table 3 confirm the trends observed for the Russian benchmark. In the non-paraphrased

setting, difflib serves as a strong baseline due to the high lexical overlap between text fragments, particularly for Equivalent spans. However, once the Equivalent fragments are paraphrased, its performance drops substantially, while proprietary LLMs become the dominant approaches. GPT-5.2 achieves the best average score in the paraphrased setting and performs best on Contradiction and Addition detection, whereas Claude Opus 4.6 slightly outperforms it on identifying paraphrased equivalents. Overall, the results demonstrate that character-level comparison is sufficient only for near-identical revisions, while paraphrase-heavy semantic diffing requires LLM-based reasoning.

| Threshold | Contr. | Add | Equiv. | Avg. |
|---|---|---|---|---|
| 25 | 51.06 | 71.11 | 93.13 | 71.76 |
| 50 | 42.55 | 72.53 | 90.51 | 68.53 |
| 75 | 29.79 | 72.53 | 59.13 | 53,81 |
| 80 | 29.79 | 72.53 | 52.73 | 51.68 |
| 85 | 25.53 | 72.53 | 41.90 | 46.65 |
| 90 | 21.28 | 72.53 | 38.10 | 43.97 |
| 95 | 21.28 | 72.53 | 33.33 | 42.38 |
| 99 | 17.00 | 59.34 | 22.68 | 33.01 |

Table 7: Performance of the GPT-5.2 model on the paraphrased LegDiff on different thresholds.

### 6.4 Threshold Sensitivity Analysis

Table 7 shows a clear inverse relation between threshold and performance: average scores decline as the threshold increases. Equivalence is the most sensitive (93.13% at threshold 25 vs 22.68% at 99), Addition is stable (plateauing at 72.53% for thresholds 50–95, dropping only at 99), and Contradiction declines gradually from 51.06% with tightening thresholds. Overall, the results at the selected threshold $\tau = 0.8$ indicate the model's approximate median performance on this dataset (average 51.68%).

## 7 Conclusion

In this work, we investigated the problem of span-level semantic comparison of legal documents. To address it, we proposed LegDiff, a novel benchmark for semantic matching in the legal domain, and developed a synthetic data generation pipeline to enable large-scale training. We conducted extensive experiments evaluating various LLM-based approaches for span-level semantic comparison. Basic character-level comparison methods performed poorly, while proprietary LLMs achieved the best results. Training on the proposed synthetic data yielded a substantial improvement in model performance that generalized across documents from different domains. These findings underscore the value of strong LLMs and task-specific synthetic data for robust span-level semantic comparison. Future work will focus on extending the approach to other languages, improving efficiency and interpretability, and enhancing integration into real-world legal and policy analysis workflows.

## Limitations

We acknowledge that the described approaches for semantic document matching and synthetic data generation have several limitations.

**Language and domain bias.** Our semantic document matching system was primarily evaluated using Russian documents from the legal domain. Broader validation across domains and languages is needed. However, we mitigated this issue by evaluating the presented methods on an English-language PropSegmEnt benchmark. Additionally, the pipelines described for generating synthetic data and constructing datasets could be applied to any language and domain with temporally varying documents. Sources such as Wikipedia and GitHub, for example, contain documents that are updated regularly.

**Small benchmark size.** LegDiff is relatively small, since high-quality span-level annotation in legal text requires careful manual work. Despite this, the proposed data construction methodology is not tied to Russian legal codes, and therefore it can be applied to any source with temporally revised documents, as noted above.

**Relying on LLM.** We acknowledge that relying on LLMs introduces variability, particularly with regard to nuanced or ambiguous phrasing. Results are not guaranteed to be reproducible. To address this limitation, we state that all LLMs were evaluated using greedy decoding with `temperature=0`, and the evaluation prompts are available on our GitHub.

## Acknowledgments

The work was supported by the grant for research centers in the field of AI provided by the Ministry of Economic Development of the Russian Federation in accordance with the agreement 000000C313925P4F0002 and the agreement with Skoltech №139-10-2025-033.

## References


Dor Bernsohn, Gil Semo, Yaron Vazana, Gila Hayat, Ben Hagag, Joel Niklaus, Rohit Saha, and Kyryl Truskovskyi. 2024. Legallens: Leveraging llms for legal violation identification in unstructured text. *arXiv preprint arXiv:2402.04335*.

Sihao Chen, Senaka Buthpitiya, Alex Fabrikant, Dan Roth, and Tal Schuster. 2023. Propsegment: A large-scale corpus for proposition-level segmentation and entailment recognition. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 8874–8893. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jyun-Yu Jiang, Mingyang Zhang, Cheng Li, Michael Bendersky, Nadav Golbandi, and Marc Najork. 2019. Semantic text matching for long-form documents. In *The World Wide Web Conference*, WWW '19, page 795–806, New York, NY, USA. Association for Computing Machinery.

A Khasianov, I Alimova, A Marchenko, G Nurhambetova, E Tutubalina, and D Zuev. 2018. Lawyer's intellectual tool for analysis of legal documents in russian. In *2018 International conference on artificial intelligence applications and innovations (IC-AIAI)*, pages 42–46. IEEE.

Yuta Koreeda and Christopher Manning. 2021. ContractNLI: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1907–1919, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Barrett Lattimer, Patrick H. Chen, Xinyuan Zhang, and Yi Yang. 2023. Fast and accurate factual inconsistency detection over long documents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1691–1703, Singapore. Association for Computational Linguistics.

Marc Marone, Orion Weller, William Fleshman, Eugene Yang, Dawn Lawrie, and Benjamin Van Durme. 2025. mmbert: A modern multilingual encoder with annealed language learning. *arXiv preprint arXiv:2509.06888*.

Sagnik Ray Choudhury, Pepa Atanasova, and Isabelle Augenstein. 2023. Explaining interactions between text spans. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12709–12730, Singapore. Association for Computational Linguistics.

Xiang Ren, Yuanhua Lv, Kuansan Wang, and Jiawei Han. 2017. Comparative document analysis for large text corpora. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 325–334, New York, NY, USA. Association for Computing Machinery.

Lucas E. Resck, Jean R. Ponciano, Luis Gustavo Nonato, and Jorge Poco. 2023. Legalvis: Exploring and inferring precedent citations in legal documents. *IEEE Trans. Vis. Comput. Graph.*, 29(6):3105–3120.

Elisei Rykov, Nikolay Ivanov, Kseniia Petrushina, Maria Bandulevich, Valentin Malykh, Vasily Konovalov, Alexander Panchenko, and Ilseyar Alimova. 2026. Sledoc: System for legal document comparison. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26)*, pages 1–6, Melbourne, VIC, Australia. ACM.

Gerrit Schumann and Jorge Marx Gómex. 2024. Detection of conflicts, contradictions and inconsistencies in regulatory documents: a literature review. In *2024 Fifth International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, pages 81–88. IEEE.

Gerrit Schumann, Jorge Marx Gómez, and Hergen Pargmann. 2026. Detection of contradictions and inconsistencies in regulatory documents using prompt-engineering.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Mengying Yuan, Wangzi Xuan, and Fei Li. 2025. Cross-document cross-lingual natural language inference via rst-enhanced graph fusion and interpretability prediction. *CoRR*, abs/2504.12324.


## A PropSegmEnt Segmentation Task Results

The segmentation results are shown in Table 8. In the fine-tuned setting, Qwen2.5-Instruct achieves the best performance on PropSegmEnt, outperforming all few-shot baselines on both Jaccard and Exact Match.

| **Method** (params) | **Mode** | **Jaccard** | **Exact** |
|---|---|---|---|
| T5-Large (0.7B) | FT | 55.50 | 32.28 |
| Qwen2.5-Instruct (32B) | FS | 37.91 | 27.59 |
| Qwen2.5-Instruct (32B) | FT | **61.69** | **37.96** |
| GPT-4o (-) | FS | 40.21 | 27.05 |

Table 8: Segmentation performance on PropSegmEnt. We report Jaccard-based span match and Exact Match. FS denotes few-shot prompting and FT denotes fine-tuning. T5-Large scores are taken from the original PropSegmEnt paper.

## B LLM Prompt used for Evaluation on LegDiff Dataset

```
[TASK DESCRIPTION]
You are a legal analyst who breaks down two complex legal paragraphs into
    minimal semantic units and establishes correspondences between them. Your
    task is to segment both paragraphs into spans, align them with each other,
    and assign a label to each pair. Spans may start and end at any point in the
    paragraph, regardless of semantic boundaries.

[CLASS DEFINITIONS]
Use three labels: equivalent, contradiction, and addition.

equivalent - a pair of spans that are either identical or paraphrased fragments
    conveying the same meaning.

contradiction - a pair of spans that refer to the same element of the statement
    (e.g., object of action, condition, basis, alternative, etc.) but contain
    different, incompatible, or mutually exclusive values/objects. If the
    contradiction is expressed at the level of a phrase or short fragment
    (object, condition, basis, alternative, etc.), extract the minimal span
    containing only the differing parts. All matching words at the beginning and
    end of the spans must be excluded and instead labeled as equivalent
    (exception: when the object at the beginning or end of the span is the
    target of the statement).

addition - a span that introduces entirely new information not present in the
    other paragraph, without contradicting or altering its core meaning, but
    rather clarifying, expanding, specifying, or adding an extra condition,
    basis, limitation, or consequence.

[IMPORTANT NOTES]
Each span must be copied exactly from the text; you must not rewrite it in any
    way. Do not provide any explanations. Output only the final JSON in the
    specified format. For addition, one of the spans may be empty.

[OUTPUT FORMAT]
[
    {
        "span_1": "<exact text span from the first paragraph>",
        "span_2": "<exact text span from the second paragraph>",
        "label": "<alignment label (equivalent / contradiction / addition)>"
    }
]
```

Figure 6: Prompt for segmentation and semantic relation classification used in our system. In prompt, we ask LLM to segment a pair of documents into aligned spans, perform reasoning on semantic relation classification, identify the label of semantic relation.

## C LLM Prompt used for PropSegmEnt Segmentation task

```
TASK:
Break down the input sentence into all possible statements formed from the
    original sentence.
The resulting statements must be assembled from the words of the original
    sentence, it is forbidden to add new ones.
Generate all possible statements.

EXAMPLES:
input: Jurassic Park (in Brazil, Jurassic Park - Parque dos Dinossauros; in
    Portugal, Jurassic Park) is a 1993 American science-fiction adventure film
    directed by Steven Spielberg and based on the eponymous book written by
    Michael Crichton.
output:
Jurassic Park book written by Michael Crichton
Jurassic Park is based on the book
Jurassic Park is directed by Steven Spielberg
Jurassic Park is a 1993 American science-fiction adventure film
Jurassic Park Portugal, Jurassic Park
Jurassic Park in Brazil Jurassic Park - Parque dos Dinossauros

input: Let's recap how things have gone: 2013 Elite Eight - The Gators came in
    as a 3-seed and the Wolverines a four.
output:
the Wolverines a four
The Gators a 3-seed
2013 Elite Eight the Wolverines
2013 Elite Eight - The Gators

input: Aubrey's head tilts and with his eyebrows furrowed, he looks as if he
    wants to say, "It's like that?"
output:
Aubrey's head tilts
Aubrey's eyebrows furrowed
Aubrey looks as if he wants to say, "It's like that?"
```

Figure 7: Prompt for the segmentation subtask in PropSegmEnt. Prompt contains several few-shots examples sampled randomly from the training subset. In prompt, we ask model to sample all possible propositions based on the examples. Only a certain few-shot examples are given, while the full prompt is available in GitHub.

## D LLM Prompt used for PropSegmEnt Entailment task

```
TASK:
Given a paragraph and a list of hypotheses, perform Natural Language Inference
    (NLI) by classifying each hypothesis into one of three categories based
    solely on its relationship to the information in the paragraph:
entailment: The hypothesis is definitely true and can be directly inferred from
    the paragraph.
contradiction: The hypothesis is definitely false and contradicts the
    information stated in the paragraph.
neutral: The hypothesis cannot be confirmed as definitively true or false based
    on the information provided in the paragraph.

Your output must be a valid JSON list of objects, where each object has the keys
    "hypothesis" and "label".

Do not write anything except of JSON in your answer.

EXAMPLE 1:
Paragraph: "Jurassic Park" is a 1993 science fiction film directed by Stephen
    Spielberg and adapted from the novel of the same name published by Michael
    Clayton in 1990.
Hypotheses:
- Jurassic Park book written by Michael Crichton
- Jurassic Park is based on the book

Output:
[{"hypothesis": "Jurassic Park book written by Michael Crichton", "reasoning":
    "The paragraph claims the novel was by 'Michael Clayton' (1990), not Michael
    Crichton, so this conflicts with the premise.", "label": "contradiction"},
{"hypothesis": "Jurassic Park is based on the book", "reasoning": "The paragraph
    states it was adapted from a novel of the same name, i.e., based on a
    book.", "label": "entailment"}]
```

Figure 8: Prompt for the entailment subtask in PropSegmEnt. Prompt contains several few-shots examples sampled randomly from the training subset. In prompt, we ask model to compare each hypothesis with the original paragraph, and perform a reasoning whether the hypothesis is entailed with the paragraph or no. Only a certain few-shot examples are given, while the full prompt is available in GitHub.