

OBLEX26: A Multi-Jurisdictional Benchmark for Structured Obligation Extraction from Anti-Money Laundering Regulations

Philip Moore

Judge Business School
University of Cambridge
pjm56@cam.ac.uk

Giovanni Bandi*

Judge Business School
University of Cambridge
g.bandi@jbs.cam.ac.uk

Tim Luka Horstmann*

Judge Business School
University of Cambridge
tlh45@cantab.ac.uk

Russell Moore

Computer Laboratory
University of Cambridge
rjm49@cam.ac.uk

Robert H. Wardrop

Judge Business School
University of Cambridge
r.wardrop@jbs.cam.ac.uk

Paula Buttery

Computer Laboratory
University of Cambridge
pjb48@cam.ac.uk

Abstract

Anti-money laundering and counter-terrorism financing (AML/CFT) regulations impose complex obligations on financial institutions. Auditing these obligations requires thousands of hours of manual review by legal experts, yet automated *structured obligation extraction* remains underexplored. In this paper we:

- 1) Introduce OBLEX26, the first multi-jurisdictional benchmarking dataset designed for evaluating structured obligation extraction from regulatory documents. The dataset contains 48 expert-annotated documents spanning 14 jurisdictions, consisting of 43 documents in English and 5 in Brazilian Portuguese and Chinese. Grounded in Hohfeldian legal theory and deontic logic, each obligation is decomposed into six components (*Holder*, *Type*, *Beneficiary*, *Antecedent*, *Consequent*, *Source / Legal Channel*) and is anchored to exact quotes, enabling verifiable provenance;
- 2) Compare four open-weight methods for structured obligation extraction that can run privately on consumer-grade hardware: token-classification encoders; a joint classification and obligation-linking architecture; zero-shot with LLMs; and fine-tuned SLMs.

Our fine-tuned SLM Qwen3.5-27B achieves a soft-match F1 score of 0.728, with 0.991 exact quote anchoring on English language data, and 0.659 F1 with 1.000 anchoring when used zero-shot on the cross-lingual data. Fine-tuned SLMs substantially outperform zero-shot LLMs and encoder-based approaches, establishing a strong baseline for automated legal compliance. OBLEX26 evaluation data is publicly released.

*Equal contribution

1- The Directorate is authorized to obtain any additional information that the Directorate deems necessary to properly carry out its analysis. In cases where a financial institution has not submitted a report under Article 70, or the Directorate's request does not relate to a report submitted by the requested financial institution, the Directorate shall request to provide the requested information only through the supervisory authority.

Figure 1: Human-annotated excerpt, using the colour coding from section 3.2. The five most common component types are present, with "source" being implicit here.

1 Introduction

Global spending on anti-money laundering (AML) compliance exceeds \$206 billion annually (Lexis-Nexis Risk Solutions, 2023).

Compliance workflows are bottle-necked because obligations are scattered across hundreds of pages of regulations and guidance. Unlike general text, legal language is constitutive: the words themselves create the legal reality, featuring long and atypical sentence structures, technical and domain-specific terms, and specially formatted documents (Kiyavitskaya et al., 2008; Katz et al., 2023). Compliance officers must manually parse this complex language to identify thousands of individual obligations, determining *who* must do *what*, under *which conditions*, and by *whose authority*. Automating extraction would scale compliance workflows and enable intelligent applications based on machine-interpretable data.

Despite progress in legal NLP, no benchmark exists for *structured obligation extraction* that captures the full normative relationship between parties, actions, situations, and legal authorities. Prior work has either focused on binary obligation detec-

tion (Zeni et al., 2015; Sleimi et al., 2018), operated on small single-jurisdiction corpora, or lacked a standardised evaluation framework.

We address this gap by introducing OBLEX26 (**O**bligation **E**xtraction **2**026). Our task formulation goes beyond traditional entity recognition or text zoning approaches: given regulatory text, the goal is to extract a set of semantic objects, each representing an obligation as a structured tuple comprising six components grounded in Hohfeldian analysis of correlative rights and duties (Hohfeld, 1917): *Holder*, *Type*, *Beneficiary*, *Antecedent*, *Consequent*, and *Source*. Each obligation is anchored to an exact span in the source text, ensuring that every semantic object has verifiable provenance; a vital characteristic for legal applications.

Our contributions are as follows:

1. We formalise *structured obligation extraction* as an information extraction task, defining a six-component semantic schema that captures the full normative structure of legal obligations.
2. We introduce OBLEX26, an expert annotated dataset comprising an English language section (43 AML/CFT regulatory documents from 12 jurisdictions), plus a cross-lingual section in Brazilian Portuguese and Chinese (5 documents from 2 jurisdictions).
3. We compare four modelling paradigms for automatically identifying structured obligations, establishing baselines for this novel task: encoders; joint models; zero-shot LLMs; and parameter-efficient fine-tuned SLMs.
4. We demonstrate cross-lingual generalisation, evaluating our best model zero-shot on unseen documents in Portuguese and Chinese.

2 Related Work

We introduce the task of *structured obligation extraction* alongside the OBLEX26 benchmark, unifying legal theory with semantic frame parsing under sequence constraints.

Extant Legal NLP Benchmarks Increased interest in Legal NLP has inspired benchmarks for legal NER (Leitner et al., 2020; Au et al., 2022), judgment prediction (Chalkidis et al., 2019; Malik et al., 2021), and contract understanding (Hendrycks et al., 2021; Koreeda and Manning, 2021), but these

benchmarks are not sufficient for evaluation of structured legal component extraction. Multi-task legal benchmarks such as LegalBench (Guha et al., 2024) and LexGLUE (Chalkidis et al., 2022) have advanced the field but the focus is on classification and entailment rather than structured extraction. In contrast, OBLEX26 provides a multi-jurisdictional benchmark that maps legal component normative relationships directly to concrete text spans.

Legal Theory and Semantic Framing Treating legal obligations as atomic units is insufficient for automated down-stream auditability. Without structural decomposition, compliance systems cannot isolate specific targets, conditions, or breach triggers within a complex clause. This work addresses this issue by defining a multi-component structure for obligations. The formal analysis of legal norms draws on Hohfeld’s framework of jural relations, specifically the correlative nature of "duty" and "right" (Hohfeld, 1917), and deontic logic (von Wright, 1951; McNamara and Van De Putte, 2021). These abstract legal concepts are then mapped to Semantic Roles (Gildea and Jurafsky, 2002) and elements of Semantic Frames (Fillmore, 1982; Baker et al., 1998). Previous Legal NLP has explored requirements engineering (Zeni et al., 2015) and regulatory compliance (Sleimi et al., 2018), but their related datasets are limited in scale and typically treat obligations as atomic units rather than decomposing them into normative components. Prior attempts at frame parsing of legal documents have been restricted to manual, template-driven software license analyses (Mandal et al., 2020).

Information Extraction Foundations Having defined structured obligations, the associated task is to extract them from regulatory documents automatically. Traditional sequence labelling tasks of Named Entity Recognition (NER) (Tjong Kim Sang and De Meulder, 2003) and Text Zoning (Teufel et al., 1999) exhibit similarities to our task. However, neither paradigm adequately captures structured obligation extraction, because target regulatory text regions are legally idiosyncratic (Malik et al., 2022), vary in frequency, feature semantic overlap, and exhibit discontinuous or incomplete coverage across the document.

Encoder vs. Decoder Approaches Token-level classification using pre-trained encoders remains the dominant paradigm for NER and information

extraction (Devlin et al., 2019; He et al., 2023; Warner et al., 2025). Joint NER and relation extraction architectures, including biaffine attention models inspired by dependency parsing (Dozat and Manning, 2017; Yu et al., 2020), have shown promise for structured extraction. Recent work has explored generative approaches to IE using instruction-tuned LLMs (Lu et al., 2022; Wang et al., 2023), demonstrating competitive performance on established benchmarks. However, the effectiveness of these paradigms for long-document, multi-component extraction from specialised legal text remains underexplored. In this work we make use of Parameter-efficient fine-tuning (PEFT) low-order rank adaptation (LoRA), (Hu et al., 2022) to efficiently fine-tune small language models (SLMs).

3 Task Definition and Annotation

3.1 Formal Task Definition

We define the task of *structured obligation extraction* as follows. Given a regulatory document $D = (w_1, w_2, \dots, w_n)$, where $w_{1..n}$ are the words composing the document, the goal is to extract the exhaustive set of semantic objects that represent the k obligations described within the document, namely $\mathcal{O} = \{o_1, o_2, \dots, o_k\}$, which may be empty. Each obligation $o_i = (q_i, \{(c_j, t_j)\}_{j=1}^{m_i})$ consists of an exact quote q_i , and a set of m_i *components* where each component consists of an exact span $c_j = \{w_k\}_{k=start_j}^{end_j}$, paired with a component type, $t_j \in \mathcal{T}$, as described in the following section.

3.2 Annotation Schema

Our annotation schema and guidelines follow Hohfeldian analysis and were created by our team of expert legal annotators. Although this study focuses on AML/CFT documents, the schema itself is domain-agnostic, describing features common to all legal obligations. The schema identifies six semantic roles for the words within an obligation, corresponding to the component type set $\mathcal{T}$ in the previous section, as follows:

- **Holder**: The legal person against whom the norm can be exercised; the party responsible for compliance.
- **Type**: The deontic modality expressing the normative force (e.g., duty, prohibition, permission).
- **Beneficiary**: The legal person who holds the norm’s correlative right.

- **Situation - Antecedent**: The set of conditions or trigger state for the norm to become active.
- **Situation - Consequent**: The state of affairs that, if brought about, satisfies the norm (the required performance).
- **Source / Legal Channel**: The specific legal instrument from which the norm derives its authority.

A link to the full annotation guidelines is given in Appendix A. Several important features follow:

- Definitional and structural clauses are excluded. In general, coverage of the text is incomplete.
- Component spans can be *shared* between obligations, e.g. the head clause of a list can be reused across all the obligations in that list.
- Every obligation must have at least one explicit component span, but generally any span can be implicit and not present in the structure. In particular, the "source" category is usually implied from the context of the document itself.
- Obligations may have more than one span of a given category.
- Spans can overlap, which means that individual words can have more than one semantic role, although this is rare in practice.

3.3 Annotation Process

The annotation team consisted of two legal experts and an expert manager as reviewer for both (see Appendix A). The team created our domain-agnostic semantic schema for legal norms, and then applied the schema to official regulatory PDF documents to produce the OBLEX26 dataset. Annotation of PDFs was carried out using the highlighting and tagging features of the free, open-source research management tool *Zotero*¹. The annotation process consists of applying the schema by highlighting spans of words using a colour codes from section 3.2. An example annotation is shown in Figure 1. Component spans are then grouped into obligation objects by tagging every related span with a common identifier (e.g. "ob1", "ob2"). These highlights and tags are exported as metadata in new PDFs for dataset preprocessing.

¹www.zotero.org

4 The OBLeX26 Dataset

4.1 Data Collection

We collected AML/CFT regulatory documents from 12 jurisdictions that officially publish in English, covering primary legislation, implementing regulations, supervisory guidance, and rulebooks. The jurisdictions were selected to represent a diverse cross-section of the global AML/CFT regulatory landscape, including major financial centres and both common-law and civil-law traditions.

While domestic AML/CFT policies are typically contained within specialised rulebooks, they broadly mirror FATF Recommendations. By targeting the specific domestic policies that mandate the enforcement of these international standards, we establish a controlled baseline for comparative analysis across these diverse jurisdictions.

4.2 Dataset Statistics

The OBLeX26 dataset consists of 43 English language documents, comprising 1767 pages, and a total of 659,756 words. These contain a total of 2842 individual semantic obligation objects, as discerned by our experts. A breakdown by jurisdiction is given in Table 1.

Statistical analysis reveals that 24.9% of all the words are annotated, giving each obligation a mean length of approximately 66 words.

Analysis of component category distribution reveals class imbalance across words and obligations:

Table 2 shows how many of the annotated words are assigned to each of the six categories of component. The vast majority (90.62%) of obligation words are contained within *consequent* (69.72%) and *antecedent* (20.90%) spans. These spans tend to contain relatively long conditional or imperative clauses, whereas *holder*, *beneficiary*, and *source* spans tend to reference an entity or instrument, usually a short name or label, and *type* is generally a short modal verb (e.g. "must").

Table 3 describes the distribution of component categories across the 2842 obligations. Classes *consequent* and *type* are ubiquitous; *holder* very common; *Antecedent* and *beneficiary* occur much less frequently; *source* is almost always implicit.

Just 9.7% of obligations cross a page-boundary in a document, meaning that the vast majority are self-contained within a single page.

Jurisdiction	Docs	Pages	Oblig.
Australia	1	374	138
Canada	1	194	224
Cayman Islands	1	80	117
EU	1	111	331
India	2	72	77
Saudi Arabia	7	181	719
Singapore	4	153	532
South Africa	3	167	117
UAE	2	160	364
UK	1	199	68
USA	19	58	120
Virgin Islands	1	18	35
Total	43	1767	2842

Table 1: Dataset statistics across 12 jurisdictions. Not shown are an additional 5 multilingual documents (Brazil, China) which are held out for zero-shot cross-lingual evaluation.

Component	Annotated Words	% of A/Words
Consequent	114508	69.72
Antecedent	34329	20.90
Holder	10575	6.44
Type	3154	1.92
Beneficiary	1470	0.90
Source	204	0.12
Total	164240	100

Table 2: Component frequency distribution across all annotated words (24.9% of all words).

4.3 Data Splits

Post-annotation, the dataset was further processed to partition it into data splits. These splits were made at document level to preserve context and prevent data leakage. The splits were stratified by jurisdiction, and the distribution of documents was heuristically chosen so that the final page count of each split met the standard ratio of 80:10:10 for train, validate and test.

4.4 Multilingual Evaluation Set

To evaluate cross-lingual generalisation, we additionally annotated 5 AML/CFT documents in Portuguese (Brazilian Central Bank regulations) and Chinese (PRC AML Law and implementation rules). These documents are strictly held out and used only for zero-shot cross-lingual evaluation.

5 Methods

We evaluate four modelling paradigms for structured obligation extraction, exploring both encoder-based and decoder-based approaches:

Component	Count	% of Oblig.
Consequent	2820	99.23
Type	2815	99.05
Holder	2378	83.67
Antecedent	1243	43.74
Beneficiary	402	14.14
Source	21	0.74

Table 3: Component class occurrence across all 2842 obligations. Consequent and Type are near-universal, while Beneficiary and Source are frequently implicit.

5.1 Baseline Models

We establish a baseline for component detection by using two dummy classifiers and the well-established BiLSTM token-classifier (Graves and Schmidhuber, 2005):

- **Random:** A random stratified classifier which assigns component labels to each token in accordance with their training set frequencies.
- **Majority:** A classifier which finds the majority component category from the training set and assigns this category to all tokens.
- **BiLSTM:** A neural token classifier with learnable word embeddings, bidirectional LSTM layer, and fully connected classification head, outputting raw logits for each component label (output dimension 7). Learning occurs across 10 epochs, and aims to minimise standard *cross-entropy loss*.

Each token is classified with a single label. This can be one of the six component categories from our semantic schema (see Section 3.2), or a seventh category ("O") denoting that the token is "outside" any component. Full 'BIO/IOB' tagging (Ramshaw and Marcus, 1995) is not used, as neighbouring spans are considered contiguous in our schema. These models operate only at the token level: no attempt is made to group runs of tokens into higher-level obligation objects.

5.2 Encoder-Based Approaches

Encoder-only transformer architectures possess several structural characteristics that are advantageous for legal information retrieval: they offer greater computational efficiency during training and lower latency during inference compared to large autoregressive alternatives (Sanh et al., 2019; Tay et al., 2022). Furthermore, because they operate via token classification and extraction rather

than text generation, they are structurally immune to generative hallucinations, ensuring that every output remains strictly anchored to a verifiable span in the source text (Ji et al., 2023). On the other hand, they are constrained by fixed-size context windows and a restricted output vocabulary limited to predefined label spaces. Encoders enrich each token with a deep, bidirectional contextual representation and automatically extract highly discriminative features (Devlin et al., 2019). This capacity should enable them to effectively detect the activation of the obligation semantic frame and precisely assign corresponding roles to each involved token (Baker et al., 1998). Consequently, we investigated two encoder configurations of increasing architectural sophistication.

5.2.1 Component Detection

Moving beyond the baseline models, we next implement an encoder-based token classifier that attempts to detect obligation frames in the input text and assign a single-label component classification to each token involved. To address the marked class imbalances discussed in Section 4.2, we use a customised *cross-entropy loss* function with losses weighted by the inverse frequency of token labels in the training set (King and Zeng, 2001). Specifically, these weights were computed utilising the 'balanced' weighting implementation from **scikit-learn** (Pedregosa et al., 2011). Our model is designed so that the transformer backbone can be switched out, allowing us to repeat the experiment with three pre-trained encoders from the BERT family:

- **RoBERTa** (Liu et al., 2019): A BERT variant with improved performance through a modified training strategy
- **Legal-BERT** (Chalkidis et al., 2020): A BERT model pre-trained on legal corpora from multiple jurisdictions.
- **ModernBERT** (Warner et al., 2025): A recently released encoder with Flash Attention 2 support and a native 8,192-token context window.

Models are trained for 10 epochs with early stopping (patience of 3 epochs), and a learning rate of 1×10^{-5} .

5.2.2 Joint Detection and Obligation Linking

Token classification identifies components but does not group them into coherent obligations. We

thus implement a *multi-task learning* (Caruana, 1997) model that attempts both tasks simultaneously. A shared ModernBERT backbone feeds context-enriched embeddings into two classification heads:

- **A Component Detection Head:** fully-connected layer with class-weighted cross-entropy loss function, returns a single class per token.
- **A Linking Head:** directly predicts a binary membership vector of up to 16 obligations in a given sample, using binary cross-entropy loss function.

The losses from the two heads are combined using blending weight $\alpha = 0.5$. Gradients from both classification heads propagate to the encoder backbone so that loss is minimised by extracting latent features that are relevant to both tasks.

The linking head assigns each component to one or more of 16 obligations per sample. However, since obligations are grouped with a simple textual tag (e.g. "ob1"), they can be arbitrarily relabelled, creating a permutation problem. This is resolved in the training set by lexically sorting the rows of the membership matrix so that an earlier occurrence of tokens conveys a lower rank. Obligations beyond 'rank 16' are omitted from training.

The model is trained for 10 epochs using learning rate of 1×10^{-5} , and a warm-up ratio of 0.1.

5.3 Decoder-Based Approaches

Autoregressive decoder-only architectures offer distinct capabilities and trade-offs for legal information retrieval. These models excel at open-ended generation, complex reasoning, and zero-shot prompting without requiring task-specific architectural modifications (Radford et al., 2019; Brown et al., 2020). Very large context windows allow them to process multi-page chunks or even complete documents in a way that encoders typically cannot. However, decoder models introduce critical vulnerabilities into deterministic legal workflows: because they rely on iterative next-token generation, they are prone to generative hallucinations, potentially fabricating fictitious results that cannot be anchored back to the source text (Ji et al., 2023). Furthermore, autoregressive decoding introduces challenging inference latency and substantial computational costs compared to encoder baselines (Tay et al., 2022). Finally, their generative flexibility raises difficulties in enforcing strictly structured output formats (e.g. JSON objects).

5.3.1 Zero-Shot LLMs

We evaluate three popular large language models in a zero-shot setting. In the interests of organisational AI sovereignty, we choose models that are open-weight and can be run privately on consumer-grade hardware:

- **DeepSeek-R1** (70B parameters)
- **GPT-OSS** (20B parameters)
- **Qwen3** (32B parameters)

Models are prompted with a summarised version of the annotation guide (see Appendices A and C) including component definitions, and a description of the required output format: a JSON array of obligations, each containing an exact quote from the source text and a components dictionary mapping component types to lists of extracted text spans. No in-context training examples were provided.

5.3.2 Fine-Tuned Small Language Models

We fine-tune a family of smaller, open-weight Qwen models of increasing size using *low rank adaptation* (LoRA) (Hu et al., 2022) with rank $r=16$ and scaling factor $\alpha=32$:

- **Qwen3-4B, Qwen3-8B, Qwen3-14B**
- **Qwen3.5-9B, Qwen3.5-27B**

LoRA fine-tuning allows the base model to be frozen, while an *adapter* containing a tiny percentage of the total neural weights is trained instead. Synthetic training examples are created by combining data from the training set with the prompt from section 5.3.1. Model adapters are then trained for 3 epochs with supervised fine-tuning, using response-only loss masking so that the loss is computed only on the model's structured output, rather than the complete conversation.

For comparison, each model is evaluated in both zero-shot and fine-tuned LoRA modes, using the same prompt as before.

5.4 Evaluation Protocol

We evaluate the baseline and component-detection models at the token level, and all other approaches using the following complementary metrics:

Token: Compare predicted per-token component labels with those from the ground truth. For tokens that are multi-labelled in the ground truth, only the first label is used. Calculate macro precision, recall and F1 score.

Approach / Model	F1	Prec.	Recall
<i>Baselines</i>			
Dummy - Majority	0.107	0.085	0.143
Dummy - Random Stratified	0.135	0.139	0.141
BiLSTM	0.506	0.664	0.409
<i>Component Detection - Encoder-Based</i>			
RoBERTa-base	0.603	0.526	0.756
ModernBERT-base	0.539	0.612	0.567
Legal-BERT	0.592	0.536	0.690

Table 4: Token-level component detection results on the OBLEX26 test set. Scores are given for successful classification of individual tokens into each component category. F1, Precision and Recall are macro.

Exact Match: Match every predicted (span, category) pair using strict string equality, and compute precision, recall, and F1 score.

Soft Match: To account for near-correct extractions, compute precision, recall and F1 using the ROUGE-L metric (Lin, 2004) at the token level between predicted and ground truth component spans. Pairs with $\text{ROUGE-L} \geq 0.5$ are considered matches. This is a commonly used threshold which detects strong structural overlap while allowing minor variations. We also report these metrics broken down by component types.

Quote Anchoring Rate: Measure the percentage of predicted obligation quotes that can be located in the source text, first by exact string match and then by fuzzy matching (Bachmann, 2025).

6 Experiments and Results

6.1 Main Results

Table 4 presents the results from our token-level experiments. All of the neural models (including the baseline BiLSTM) outperform the dummy baselines. On average, their F1 score is +0.425 better than random (0.135). The best precision (0.664) is achieved by the BiLSTM, although its corresponding recall (0.409) is the worst of the neural models. Of the three encoders, **RoBERTa** achieves the best overall result, with both the highest recall (0.756) and overall F1 score (0.603). **Legal-BERT** shows very similar performance, and **ModernBERT** lags behind, albeit with the highest precision (0.612) of the three.

Table 5 shows the results from our experiments with the more advanced obligation-grouping models. The **joint encoder** is the least successful model in this group, achieving an F1 score of only 0.264,

even when using soft matching. The three zero-shot LLMs all exhibit similar levels of performance to each other. Token F1 scores sit in the range 0.700-0.734. When exact matching is used to tie spans back to components, all three models fare poorly: F1 scores sit in the range 0.140-0.198, with **GPT-OSS 20B** doing best across all exact metrics. Soft matching yields better scores, with **Qwen3-32B** achieving the best F1 (0.528). Interestingly, **Deepseek-R1-70B** exhibits better recall (0.734) than precision (0.387), whereas the opposite is true for the other two models. Quote anchoring performance is broadly similar across all three models: between 69.2-78.9% of predicted obligations are tied back to exact strings in the original document. With the less strict fuzzy matching, success becomes near universal for all three models (97.2-99.9%).

When we evaluate the SLMs in a zero-shot setting, we observe similar patterns to those seen for the LLMs: token F1 scores sit in the range 0.710-0.736; exact F1 scores are low (0.164-0.219). Soft match F1 scores are in a similar range (0.459-0.547), although for the SLMs these are largely supported by higher recall scores (0.652-0.774). The **Qwen3-14B** model performs strongly across all metrics, while **Qwen3.5-27B** has the highest recall (0.774) and best quote anchoring (0.892) of any zero-shot model.

Finally, we evaluate SLMs that have been fine-tuned on page-level training data. This dramatically improves the exact F1 scores, with **Qwen3.5-27B** scoring 0.600 compared to 0.198 in zero-shot. In fact, fine-tuned **Qwen3.5-27B** scores the highest across all metrics for all models in any setting, with a Soft F1 of 0.728, strong precision (0.698) and recall (0.760) and a exact quote anchoring score of 0.991. Interestingly, two of the models, **Qwen3-8B**, **Qwen3-14B** are *weakened* by fine-tuning, exhibiting lower precision and recall compared to zero-shot mode (discussed in section 7).

6.2 Per-Component Analysis

Table 6 breaks down F1 scores by component type, comparing the best encoder (**RoBERTa**) with the best decoder (**fine-tuned Qwen3.5-27B**). Both models score most highly for the most frequently encountered component categories (*Consequent*, *Type* and *Holder*), with performance tailing off for rare categories (*Beneficiary* and *Source*). See Appendix D for full breakdown.

Approach	Model	Tok. F1	Exact F1	Soft F1	Precision	Recall	Ex. QA	Fz. QA
<i>Encoder-Based</i>								
Joint Detect+Link	ModernBERT+Linker	0.468	0.080	0.264	0.255	0.267	0.082	0.204
<i>Zero-Shot LLMs</i>								
Zero-Shot	DeepSeek-R1 70B	0.703	0.193	0.507	0.387	0.734	0.773	0.999
Zero-Shot	GPT-OSS 20B	0.734	0.198	0.512	0.686	0.409	0.789	0.972
Zero-Shot	Qwen3 32B AWQ	0.700	0.140	0.528	0.716	0.418	0.692	0.996
<i>Zero-Shot SLMs</i>								
Zero-Shot	Qwen3-4B	0.730	0.164	0.459	0.354	0.652	0.698	0.999
Zero-Shot	Qwen3-8B	0.722	0.185	0.483	0.374	0.679	0.758	0.992
Zero-Shot	Qwen3-14B	0.736	0.219	0.547	0.442	0.717	0.739	0.999
Zero-Shot	Qwen3.5-9B	0.710	0.189	0.498	0.386	0.704	0.818	0.996
Zero-Shot	Qwen3.5-27B	0.719	0.198	0.506	0.376	0.774	0.892	0.999
<i>Fine-Tuned SLMs</i>								
Fine-Tuned	Qwen3-4B	0.828	0.460	0.622	0.674	0.579	0.834	1.000
Fine-Tuned	Qwen3-8B	0.784	0.457	0.435	0.324	0.661	0.716	0.996
Fine-Tuned	Qwen3-14B	0.810	0.489	0.434	0.326	0.646	0.959	1.000
Fine-Tuned	Qwen3.5-9B	0.825	0.559	0.693	0.639	0.756	0.979	1.000
Fine-Tuned	Qwen3.5-27B	0.842	0.600	0.728	0.698	0.760	0.991	1.000
<i>Multilingual (English fine-tuned)</i>								
Zero-shot	Qwen3.5-27B	0.940	0.574	0.659	0.652	0.666	1.000	1.000

Table 5: Main results. Precision and recall correspond to Soft matches. "Ex. QA" and "Fz. QA" are exact and fuzzy quote anchoring respectively.

Component	Freq.	Enc. F1	Dec. F1
Consequent	0.992	0.739	0.696
Type	0.991	0.620	0.809
Holder	0.837	0.677	0.798
Antecedent	0.437	0.618	0.639
Beneficiary	0.141	0.361	0.481
Source	0.007	0.398	0.000

Table 6: Per-component F1 for best encoder (RoBERTa) and best decoder (fine-tuned Qwen3.5-27B).

6.3 Multilingual Zero-Shot Transfer

We assess zero-shot cross-lingual transfer using our fine-tuned **Qwen3.5-27B** on held-out Portuguese and Chinese AML documents, with no multilingual fine-tuning. Table 5 shows similar performance to English, with a soft F1 score of 0.659, and quote anchoring of 1.000.

7 Discussion

Encoders vs. Decoders. Our results clearly favour decoder-based approaches over encoders in structured legal obligation extraction. Our component-only encoders were able to detect and classify the six semantic roles, but the joint model failed to successfully group spans: unstable behaviour from the linking head dragging down overall performance. A redesign of this head is considered for future work. With careful grounding through quote anchoring, we have shown that

generative decoders can produce structured outputs that faithfully map back to the source text.

Zero-shot vs. Fine-tuned Behaviour. All models evaluated in zero-shot mode suffered from two common difficulties. Firstly, their exact match F1 scores were poor. Even though they were prompted to quote text *verbatim*, they generally failed to do so, and often removed hyphenation or punctuation from the original text. Secondly, being unaware of the relative frequencies of implicit components, they returned far too many 'source' and 'beneficiary' components, which dragged down their precision even under soft match conditions. Fine-tuned models, with the benefit of many worked examples, were much better at returning text in the preferred verbatim style, and also better at discriminating whether components should be included at all. An interesting future study would be to explore behaviour in a few-shot setting. The weakening of two of the SLMs (**Qwen3-8B**, **Qwen3-14B**) under fine-tuning conditions stems from a collapse in their precision. Careful study of their raw responses reveals this is caused by massive repetition of outputs, leading to inflated false positives. It is possible that this is caused by a 'reasoning loop' in the larger Qwen3 models (Pipis et al., 2026), and we hope to address this in follow-up work, including by analysing 'chains of thought'.

The Challenge of Rare Components. The ‘Beneficiary’ and ‘Source’ components are frequently implicit in regulatory text. Class weights help to address this for encoder models. ‘Source’ occurs so infrequently that its inclusion actually confuses zero-shot decoder models and harms their precision. The best-performing fine-tuned SLMs do not predict ‘source’ at all, possibly because they never reach the required confidence to do so. Strategies to handle these rare component categories warrant further investigation.

Practical Implications. For compliance applications, even imperfect extraction is valuable as a pre-filtering step that reduces manual review effort. The quote anchoring mechanism ensures that all extractions are traceable, enabling compliance officers to verify and correct model outputs. Semi-automated extraction with human-in-the-loop validation represents the most realistic near-term deployment scenario.

8 Conclusion

We introduced OBLEX26, the first multi-jurisdictional benchmark for structured legal obligation extraction from AML/CFT regulations. Our dataset comprises 43 expert-annotated documents across 12 jurisdictions, with obligations decomposed into six components grounded in deontic logic. A systematic comparison of four modelling paradigms reveals that fine-tuned small language models outperform a range of popular LLMs in a zero-shot setting, while encoder models are able to detect the components of our schema, but as yet unable to aggregate them successfully into semantic objects. We release the evaluation data to establish a shared benchmark for the legal NLP community, with the aim of advancing research at the intersection of natural language processing and legal compliance.

Limitations

Our work has several limitations. First, the dataset covers a single regulatory domain (AML/CFT); while the annotation schema is domain-agnostic and grounded in general deontic logic, generalisation to other regulatory domains (e.g., financial services conduct rules, data protection, environmental regulations) remains to be demonstrated. Second, the dataset is predominantly English, with the multilingual evaluation set limited to

Portuguese and Chinese. Broader multilingual coverage would strengthen claims about cross-lingual transfer. Third, our annotation was performed by a small team of legal experts; while we conducted quality checks to identify obligations with missing components, formal inter-annotator agreement metrics would further validate annotation consistency. Finally, the current evaluation metrics do not fully capture the practical utility of extracted obligations for downstream compliance tasks. Extrinsic evaluation in real compliance workflows would provide stronger evidence of practical value.

Ethics Statement

The regulatory documents used in this study are publicly available legislation and regulatory guidance published by government authorities. No personal data or confidential information is included in the dataset. Our work aims to support compliance professionals and should not be used as a substitute for qualified legal advice. The automated extraction of legal obligations carries risks of misinterpretation; we emphasise the importance of human oversight in any practical deployment.

Acknowledgments

The authors gratefully acknowledge the financial support of our sponsors, who have elected to remain anonymous for reasons of regulatory independence.

Separately, the authors thank Mr Chao Zhou and Mr Fernando Barroso for their support and expertise in legal annotation.

Finally, the authors express their gratitude to the anonymous reviewers for their insightful comments, which helped to improve and strengthen this paper.

References

- Ting Wai Terence Au, Vasileios Lamos, and Ingemar Cox. 2022. [E-NER — an annotated named entity recognition corpus of legal texts](#). In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 246–255.
- Max Bachmann. 2025. [rapidfuzz/rapidfuzz: Release 3.14.3](#).
- Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. [The Berkeley FrameNet project](#). In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90,

- Montreal, Quebec, Canada. Association for Computational Linguistics.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Rich Caruana. 1997. [Multi-task learning](#). *Machine Learning*, 28(1):41–75.
- Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. [Neural legal judgment prediction in English](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323.
- 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.
- Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022. [LexGLUE: A benchmark dataset for legal language understanding in English](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 4310–4330.
- 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*, pages 4171–4186.
- Timothy Dozat and Christopher D. Manning. 2017. [Deep biaffine attention for neural dependency parsing](#). In *International Conference on Learning Representations*.
- Charles Fillmore. 1982. Frame semantics. *Linguistics in the Morning Calm*. Hanshin, Seoul, pages 111–137.
- Daniel Gildea and Dan Jurafsky. 2002. [Automatic labeling of semantic roles](#). *Computational linguistics*, 28(3):245–288.
- Alex Graves and Jürgen Schmidhuber. 2005. [Framewise phoneme classification with bidirectional lstm and other neural network architectures](#). *Neural Networks*, 18(5-6):602–610.
- Neel Guha, Julian Nyarko, Daniel E Ho, Christopher Ré, Adam Chilton, Haresh Narasimhan, and 1 others. 2024. LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in Neural Information Processing Systems*, 36.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. [DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing](#). In *The Eleventh International Conference on Learning Representations*.
- Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. [CUAD: An expert-annotated NLP dataset for legal contract review](#). In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.
- Wesley Newcomb Hohfeld. 1917. Fundamental legal conceptions as applied in judicial reasoning. *Yale Law Journal*, 26(8):710–770.
- 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 *International Conference on Learning Representations*.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. [Survey of hallucination in natural language generation](#). *ACM Computing Surveys*, 55(12):1–38.
- Daniel Martin Katz, Dirk Hartung, Lauritz Gerlach, Abhik Jana, and Michael J. Bommarito II. 2023. [Natural Language Processing in the Legal Domain](#). *arXiv preprint*. ArXiv:2302.12039 [cs].
- Gary King and Langche Zeng. 2001. [Logistic regression in rare events data](#). *Political Analysis*, 9(2):137–163.
- Nadzeya Kiyavitskaya, Nicola Zeni, Travis D Breaux, Annie I Antón, James R Cordy, Luisa Mich, and John Mylopoulos. 2008. [Automating the extraction of rights and obligations for regulatory compliance](#). In *International conference on conceptual modeling*, pages 154–168. Springer.
- 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.
- Elena Leitner, Georg Rehm, and Julián Moreno-Schneider. 2020. A dataset of German legal documents for named entity recognition. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4478–4485.
- LexisNexis Risk Solutions. 2023. True cost of financial crime compliance global report. Available at <https://risk.lexisnexis.com/global/en/insights-resources/research/true-cost-of-financial-crime-compliance-study-global-report>.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [RoBERTa: A Robustly Optimized BERT Pretraining Approach](#). *arXiv preprint*. ArXiv:1907.11692 [cs.CL].
- Yaojie Lu, Qian Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. [Unified structure generation for universal information extraction](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 5755–5772.
- Vijit Malik, Rishabh Sanjay, Shouvik Kumar Guha, Angshuman Hazarika, Shubham Kumar Nigam, Arnab Bhattacharya, and Ashutosh Modi. 2022. Semantic segmentation of legal documents via rhetorical roles. In *Proceedings of the Natural Legal Language Processing Workshop (NLLP)*, pages 131–145.
- Vijit Malik, Rishabh Sanjay, Shubham Kumar Nigam, Kripabandhu Ghosh, Shouvik Kumar Guha, Arnab Bhattacharya, and Ashutosh Modi. 2021. [ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4046–4062, Online. Association for Computational Linguistics.
- Sayonnha Mandal, Robin Gandhi, and Harvey Siy. 2020. [Modular norm models: practical representation and analysis of contractual rights and obligations](#). *Requirements Engineering*, 25(3):383–412.
- Paul McNamara and Frederik Van De Putte. 2021. Deontic Logic. In *The Stanford Encyclopedia of Philosophy*, Spring 2021 edition. Metaphysics Research Lab, Stanford University.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Charilaos Pipis, Shivam Garg, Vasilis Kontonis, Vaishnavi Shrivastava, Akshay Krishnamurthy, and Dimitris Papailiopoulos. 2026. [Wait, wait, wait... why do reasoning models loop?](#) In *Forty-third International Conference on Machine Learning*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Lance Ramshaw and Mitch Marcus. 1995. [Text chunking using transformation-based learning](#). In *Third Workshop on Very Large Corpora*.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. [DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter](#). In *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing @ NeurIPS 2019*.
- Amin Sleimi, Marcello Ceci, Mikel I. Bagüés, and Miren Visse. 2018. [Automated extraction of semantic legal metadata using natural language processing](#). *Proceedings of the IEEE 26th International Requirements Engineering Conference*, pages 124–135.
- Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey. *ACM Computing Surveys (CSUR)*, 55(6):1–28.
- Simone Teufel, Jean Carletta, and Marc Moens. 1999. [An annotation scheme for discourse-level argumentation in research articles](#). In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 110–117, Bergen, Norway. Association for Computational Linguistics.
- Erik F. Tjong Kim Sang and Fien De Meulder. 2003. [Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition](#). In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.
- Georg Henrik von Wright. 1951. [Deontic logic](#). *Mind*, 60(237):1–15.
- Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023. [Instructuie: Multi-task instruction tuning for unified information extraction](#). *Preprint*, arXiv:2304.08085.
- Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. [Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria. Association for Computational Linguistics.
- Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. [Named entity recognition as dependency parsing](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476.
- Nicola Zeni, Nadzeya Kiyavitskaya, Luisa Mich, James R Cordy, and John Mylopoulos. 2015. [GaiusT: supporting the extraction of rights and obligations for regulatory compliance](#). *Requirements engineering*, 20(1):1–22.

A Annotation Schema

A.1 Annotation Team

The legal annotation was undertaken by a team of three financial regulation experts affiliated with the Regulatory Genome Project (RGP) at Judge Business School, University of Cambridge. Collectively, the team combines doctoral-level legal research, postgraduate qualifications in both law and finance, and professional regulatory experience across several jurisdictions. Dr Bandi (PhD, Durham University; MSc in Finance and Banking, Bocconi University), Executive Director of the RGP, coordinated the exercise and acted as reviewer. Mr Zhou (PhD Candidate, University of Manchester; LLM, Queen Mary University of London, MSc Economic Law, Sciences Po) and Mr Barroso (MSc in Law and Finance, University of Oxford, LLM, University of Camerino), both Regulatory Analysts at the RGP, served as the primary annotators. The team’s combined grounding in the civil law and common law traditions afforded a broader interpretive basis for the regulatory texts under examination.

A.2 Annotation Guide

The Annotation Guide in PDF format can be found at the project’s public Github repository, here: <https://github.com/pjm56/oblex26/blob/main/2026%2009%2023%20-%20Annotation%20Guidelines%20FINAL.pdf>

Figure 2 gives a visual guide to the colour-coding scheme that was used during the annotation process in Zotero.

Colour-code: *Holder*, *Type*, *Beneficiary*, *Antecedent*, *Consequent*, *Source*.

Figure 2: The Zotero colour-coding for the six possible annotation highlights.

B Dataset and Examples

The **test data samples** are available here for academic use:

<https://github.com/pjm56/oblex26/blob/main/oblex26-test-data.zip>

See Figure 3 for an example of two obligations that share components and overlap (in this case due to being part of a list with a common head).

C Implementation Details

See Table 8 for hyperparameter settings used across all approaches.

C.1 Prompts

See Table 7 for the standard system prompt.

D Full Per-Component Results

See Table 9 for a breakdown of per component soft match performance.

You are a legal norms extraction system.

TASK:

Identify all legal norms in the text and mark their components with the following six labels.

LABELS:

- ANTECEDENT: The "IF" part: conditions that activate the norm.
- CONSEQUENT: The "THEN" part: required performance / outcome that satisfies the norm (can be prescriptive or remedial, depending on the instrument).
- HOLDER: The person/entity against whom the right is exercised / who must comply (=Duty-bearer).
- BENEFICIARY: The person/entity holding the right/benefit (=Right-holder).
- TYPE: The norm's deontic modality (what kind of prescription it is), e.g., duty or right.
- SOURCE: Where the norm gets legal authority (instrument + provision context).

OUTPUT FORMAT:

Return a JSON object with a list of "obligations".

For each obligation, you MUST provide:

1. "quote": The EXACT sentence or clause from the text containing the norm.
2. "components": Extract specific EXACT substrings from the "quote" for roles like HOLDER, TYPE, etc.

RULES:

- The "quote" must be an EXACT copy from the source text.
- Each obligation gets its own entry with its own "quote". If a sentence contains multiple distinct obligations, each entry should include the full sentence or clause as the "quote".
- DO NOT PARAPHRASE or SUMMARIZE the "quote" nor the components. They must be VERBATIM from the text!
- If a component is not explicitly present in the "quote", omit it from the output entirely.

JSON SCHEMA:

```
{
  "obligations": [
    {
      "quote": "<exact sentence or clause from the text>",
      "components": {
        "TYPE": ["<deontic modality like 'shall', 'must', 'may'>"],
        "HOLDER": ["<entity who must comply>"],
        "BENEFICIARY": ["<entity who benefits (if applicable)>"],
        "ANTECEDENT": ["<conditions that activate the norm (if applicable)>"],
        "CONSEQUENT": ["<required action or outcome>"],
        "SOURCE": ["<legal authority reference (if applicable)>"]
      }
    }
  ]
}
```

If no obligations are found in the text, return: {"obligations": []}

Table 7: The standard system prompt summarising the annotation guidance and output schema.

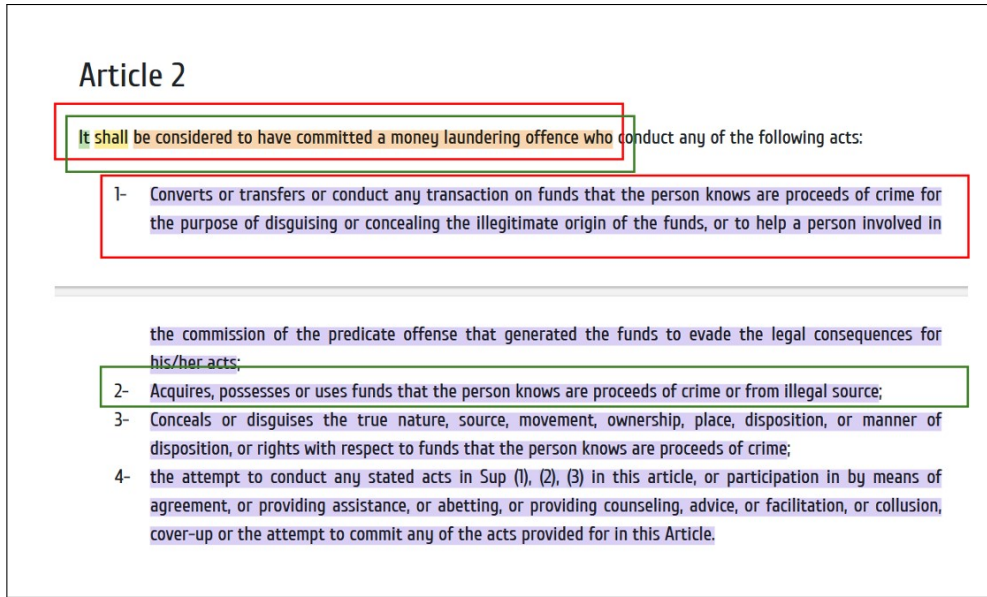


Figure 3: Two obligations (red box and green box) that share a common set of components for the header of the list, and therefore overlap.

Hyperparameter	Value
<i>Encoder (Token Classification)</i>	
Learning rate	1×10^{-5}
Epochs	10 (early stop, patience 3)
Max sequence length	512
<i>Fine-Tuned SLM (LoRA)</i>	
LoRA rank (r)	16
LoRA alpha (α)	32
Learning rate	1×10^{-5}
Epochs	3
Batch size	1
Gradient accumulation steps	8
Max sequence length	32,768
Max new tokens	16,384
Quantisation	4-bit (bitsandbytes)
<i>Joint Architecture</i>	
Feature dim	300
Dropout	0.1
Warmup ratio	0.1
Learning rate	1×10^{-5}
Linking loss weight	0.5

Table 8: Hyperparameters for all modelling approaches.

Model	Approach	Metric	Hold.	Ante.	Type	Bene.	Cons.	Sour.
Deepseek-R1-70B	Zeroshot	F1	0.568	0.413	0.585	0.046	0.552	0.000
		Pre	0.416	0.362	0.446	0.034	0.463	0.000
		Rec	0.898	0.482	0.852	0.071	0.684	0.000
GPT-OSS-20B	Zeroshot	F1	0.626	0.525	0.609	0.062	0.553	0.000
		Pre	0.504	0.442	0.501	0.046	0.517	0.000
		Rec	0.826	0.648	0.778	0.095	0.594	0.000
Qwen3-32B	Zeroshot	F1	0.591	0.430	0.583	0.031	0.596	0.000
		Pre	0.448	0.410	0.454	0.043	0.524	0.000
		Rec	0.869	0.451	0.816	0.024	0.691	0.000
Qwen3-4B	Zeroshot	F1	0.596	0.403	0.611	0.024	0.523	0.000
		Pre	0.457	0.368	0.486	0.016	0.527	0.000
		Rec	0.855	0.446	0.823	0.048	0.519	0.000
Qwen3-8B	Zeroshot	F1	0.571	0.352	0.562	0.000	0.542	0.000
		Pre	0.426	0.289	0.437	0.000	0.484	0.000
		Rec	0.863	0.451	0.787	0.000	0.616	0.000
Qwen3-14B	Zeroshot	F1	0.633	0.370	0.603	0.028	0.574	0.000
		Pre	0.482	0.324	0.475	0.033	0.507	0.000
		Rec	0.920	0.430	0.828	0.024	0.661	0.000
Qwen3.5-9B	Zeroshot	F1	0.608	0.430	0.582	0.015	0.509	0.000
		Pre	0.464	0.316	0.456	0.011	0.450	0.000
		Rec	0.882	0.674	0.804	0.024	0.586	0.000
Qwen3.5-27B	Zeroshot	F1	0.620	0.465	0.611	0.054	0.561	0.004
		Pre	0.465	0.326	0.476	0.038	0.488	0.002
		Rec	0.933	0.808	0.854	0.095	0.659	0.200
Qwen3-4B	Fine-tuned	F1	0.673	0.450	0.745	0.364	0.552	0.000
		Pre	0.639	0.677	0.749	0.769	0.634	0.000
		Rec	0.710	0.337	0.742	0.238	0.489	0.000
Qwen3-8B	Fined-tuned	F1	0.451	0.416	0.464	0.175	0.414	0.000
		Pre	0.311	0.391	0.325	0.333	0.324	0.000
		Rec	0.818	0.446	0.809	0.119	0.574	0.000
Qwen3-14B	Fine-tuned	F1	0.451	0.416	0.464	0.175	0.414	0.000
		Pre	0.311	0.391	0.325	0.333	0.324	0.000
		Rec	0.818	0.446	0.809	0.119	0.574	0.000
Qwen3.5-9B	Fine-tuned	F1	0.745	0.620	0.782	0.045	0.635	0.000
		Pre	0.635	0.582	0.688	0.500	0.621	0.000
		Rec	0.901	0.663	0.907	0.024	0.649	0.000
Qwen3.5-27B	Fine-tuned	F1	0.798	0.639	0.809	0.481	0.670	0.000
		Pre	0.753	0.596	0.736	0.403	0.696	0.000
		Rec	0.850	0.689	0.900	0.595	0.646	0.000

Table 9: Model performance comparison for soft match across different component types