Interpretable Legal Similarity: From Embeddings to Obligations

Adam Roegiest
adam@roegiest.com
Zuva
Toronto, Canada

Johanne R. Trippas
j.trippas@rmit.edu.au
RMIT University
Melbourne, Australia

Abstract

Lawyers routinely compare contracts and their constituent clauses to help clients evaluate potential acquisitions, assess risks when selling a business, or audit agreements in response to regulatory change. Clause retrieval is often supported by embedding-based similarity search, which can efficiently surface clauses that match a desired template. However, prior work suggests that using embedding similarity for review triage may lead lawyers to ignore problematic clause language.

In this work, we examine 10 clause types and demonstrate that ranking clauses based on shared and unique legal obligations, using one or more prompts, can align as closely or more closely with human lawyer judgments than embedding-based similarity, while offering interpretable outputs that embeddings do not. We highlight that prompt-based methods can produce more discriminative similarity scores than embeddings, which may aid in setting review thresholds. Finally, we discuss extensions to our prompt-based methodology that would allow lawyers to customize similarity criteria for specific use cases, a flexibility that embedding-based approaches do not readily support.

CCS Concepts

• **Information systems** → **Document representation**.

Keywords

embedding, semantic similarity, legal, contract

ACM Reference Format:
Adam Roegiest and Johanne R. Trippas. 2026. Interpretable Legal Similarity: From Embeddings to Obligations. In *Proceedings of the 2026 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR) (ICTIR '26), July 25, 2026, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3805713.3820406

1 Introduction

Contract review is central to legal operations, particularly during regulatory change [9, 10] or mergers and acquisitions [15, 21, 29, 34], where large-scale clause analysis is required. In such settings, lawyers must quickly identify clauses[1] whose obligations may affect compliance or risk the transaction. For example, recent regulatory changes [10] may require a company to review its contracts to ensure that they are compliant with the law and amend contracts when they are not.

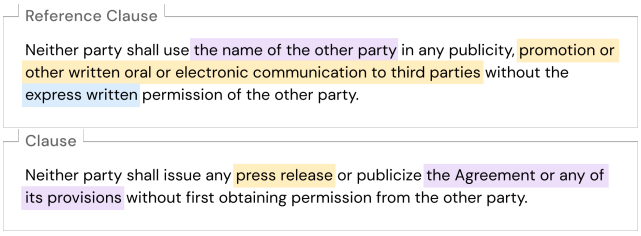

Figure 1: A version of a *semantic redline* that highlights different obligations between a reference clause and a different clause. Shared colours indicate related legal concepts.

Clauses can be efficiently and effectively ranked by similarity to a reference clause using embeddings [8]. Ideally, this "semantic" similarity would reflect legal equivalence (e.g., where two contract clauses that are legally equivalent have high "semantic" similarity irrespective of specific wording). However, legal language reflects specific intent and interpretation, and minor variations can substantially alter meaning. Prior work by Donnelly and Roegiest [8] shows that embeddings often fail to capture such nuance, where a negated clause is found to consistently be more similar to a reference clause than an equivalent rephrasing. While illustrative, Donnelly and Roegiest's findings do not reflect realistic review workflows as they were concerned with artificially constructed clauses.

To evaluate embeddings in a more practical setting, we collected lawyer-provided similarity rankings across 10 clause types, where candidate clauses are ranked relative to a reference clause. We find that embeddings reasonably approximate these rankings but lack interpretability, limiting their utility in practice.

Lawyers use redlines (i.e., edit-distance-based comparison) to conduct clause comparisons, but redlines fail to capture semantically equivalent language. We call a potential solution to this problem a *semantic redline* (Figure 1), which seeks to identify the legal concepts that differ between clauses rather than focusing solely on lexical differences. As a first step, we evaluate three prompt-based approaches inspired by recent work on re-ranking [1, 26, 47], two that are pairwise and the other listwise, that identify overlapping and distinct obligations in clauses to compute Jaccard similarity or provide a rationale for rank orders. These outputs are not a semantic redline, but could provide the basis for one.

In this work, we show that prompt-based methods can produce rankings that align as closely or more closely with the lawyer's judgments than embeddings and provide outputs that support efficient, interpretable triage, though no single method dominates. Using an additional clause type, the lawyer echoed these results, preferring prompt-based approaches and finding the outputs more useful than

[1] From [8], "a clause defines the rights, obligations, and requirements between parties pertaining to a specific concept (e.g., confidentiality)."

ICTIR '26, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-2600-2/2026/07
https://doi.org/10.1145/3805713.3820406

opaque embedding-based ranking. While embeddings provide a useful first-pass ranking, our prompt-based approaches offer comparable or better alignment, are more amenable to customization, and provide human-interpretable outputs.

2 Background

Understanding how to represent legal documents has been an active area of research [4, 8, 20, 30, 35, 42], especially in the context of legal case retrieval and judgment prediction research that has sought to leverage structural components of cases to enable better alignment to legal reasoning [3, 16–19, 22, 38, 41, 43]. At the same time, explainable IR seeks to make retrieval decisions transparent to users [2, 28, 44], especially in professional domains [27, 36, 45]. Building on this foundation, we decompose contract clauses into their obligations and measure similarity based on obligation overlap. This obligation-level comparison provides interpretable rankings that facilitate customization to a lawyer's needs more easily than fine-tuning embedding models would.

3 Methodology

This section provides high-level methodological details before subsequent sections detail the different similarity approaches. All code, data, and prompts are available online.[2]

3.1 Clause Dataset

To foster replicability and avoid using confidential business contracts, we sourced our contract clauses from publicly available contracts.[3] Rather than synthetically modify clauses [8], which requires substantial legal expertise and may reduce ecological validity, we used pre-identified clause types and variants offered by Justia.[4] Justia organizes clause types into similarity groups that form the basis of our experiments. While it is not disclosed how similarity is determined, manual inspection determined that the clauses appeared to be valid variations of each other (i.e., pairs of clauses could be modified versions of each other). From the available clause types, ten were selected (Table 1) that would be familiar to our assessor and had a similarity group with sufficient within-group variation.

3.2 Lawyer Assessment

We used a single lawyer, formerly at a top New York law firm and experienced in contract review, to rank clauses by similarity to a reference clause. Our use of a single assessor mirrors the real world, where there is usually a single senior lawyer setting a company's position on legal matters (i.e., the general counsel). While lawyers have been noted to disagree on contractual concepts [12, 31], existing research [23, 40] indicates that the observed outcomes are likely to be consistent, even if absolute scores differ, should a different lawyer have been used.

For each clause type, the lawyer was provided a spreadsheet with the reference clause and 10–12 candidate clauses from the same Justia similarity group.[5] The lawyer was instructed to treat named entities as equivalent where contextually appropriate (e.g., "Company A" vs. "Company B") to avoid "surface" level differences that would not be present if the clauses were from the same source company. The lawyer exercised their professional judgment in collapsing such occurrences and not applying it arbitrarily to any entities, such as related defined terms (i.e., "Buyer" and "Seller" would not be collapsed). We stress that, in a real dataset, this would not be necessary, since contracts would derive from the same business. During their assessment, ties were permitted when the lawyer determined that differences between a set of clauses and the reference clause had equal magnitude (i.e., they were equivalently similar to the reference, but not necessarily to each other). This review process took 20–30 minutes per clause group and each had at least one set of ties.

3.3 Evaluation

Since the lawyer's assessments are an ideal (preference) ranking, we evaluate agreement using Rank-Biased Overlap (RBO) [6], which computes top-weighted agreement between two rankings (i.e., 1 indicates a perfect agreement and 0 indicates disagreement). As rankings contain ties, we compute the expected RBO to account for them [7]. To assess how well methods reproduce rank ties, we report Adjusted Mutual Information (AMI) [39] between the lawyer's ranking and each method. AMI measures how well a method's grouping of tied clauses into sets aligns with the lawyer's after being corrected for chance. For further analysis, we compute per-clause rank differences between the lawyer's ranking and each method. Finally, we analyze changes in pairwise similarity scores to understand how each approach structures the similarity space.

3.4 Models

For embedding-based similarity, we use OpenAI's *text-embedding-3-large*, following prior work [8], and Google's *gemini-embedding-001* as our embedding baselines. Prompt-based approaches use *gpt-4.1* and *gpt-5.2*, representing non-reasoning and reasoning large language model configurations. While model choice may influence results (Section 6.1), our objective is to compare embeddings with explainable prompt-based alternatives rather than optimize per-clause performance, which would require task- and lawyer-specific details.

4 Similarity Methods

We evaluate embedding-based, pairwise obligation-matching, and listwise ranking approaches. Note that high-level prompt details can be found in Appendix A with full prompts available as part of the code provided in the online resource.

4.1 Embeddings (E)

Given the relatively short length of the clauses and prior findings [8], we embedded the entirety of each clause using OpenAI and Google embedding APIs. While a legal-tuned baseline would be desirable, Donnelly and Roegiest [8] found that such models were less effective and introduced additional complexity due to their sentence-based nature (i.e., aligning sentences between clauses is problematic), and so we omit them. The cosine similarity between the reference embedding and each candidate embedding was used

[2] https://github.com/zuvaai/science/tree/master/ICTIR2026

[3] Originally sourced from SEC's EDGAR repository (https://www.sec.gov/search-filings) but provided by Justia (https://justia.com/).

[4] https://contracts.justia.com/

[5] Some clause types used 12 candidates due to (first author perceived) interesting variations in language.

Clause Type	Description	Clause Type	Description
(T1) Assignment	Describes how one party can transfer their rights or obligations under the contract to another party.	(T6) Non-Compete	Restricts a party from working for or starting a competing business.
(T2) Confidentiality	Requires specified parties to keep private certain information that is shared under the contract.	(T7) Non-Disparagement	Prevents specified parties from making negative or harmful statements about the other parties.
(T3) Duration	States how long the contract remains in effect before it ends or must be renewed.	(T8) Notice	Describes how official communications (e.g., late payment) must be delivered.
(T4) Force Majeure	Specifies what happens if unexpected events (e.g., war, flood, pandemics) prevent a party from fulfilling obligations.	(T9) Publicity	Conditions under which either party can publicly mention or advertise the relationship/agreement/etc.
(T5) Indemnity	Specifies whether a party agrees to cover another party's losses or legal costs if specified problems arise.	(T10) Termination	The conditions under which an employment relationship can be ended by either side.

Table 1: The ten clause types (T1–10), with a simple description and shorthand identifier, that were used in our similarity ranking experiments.

to produce the ranking. OpenAI embeddings are reported as **O** and Google embeddings as **G**. These systems represent accessible and efficient baselines available to system builders.

4.2 Pairwise Similarity

These approaches compare clauses by decomposing them into obligations and aligning those obligations across clause pairs. For each pair, the model identifies four sets: equivalent obligations, opposite obligations (e.g., "you *may* assign this" vs. "you *may not* assign this"), obligations present only in the reference clause, and obligations present only in the candidate clause. While there are no "opposite" obligations in our dataset, we kept this as we piloted these methods on the Donnelly and Roegiest dataset that contained them [8]. Similarity is computed using Jaccard overlap, treating equivalent obligations as the intersection and all identified obligations as the union. Accordingly, clause pairs with fewer equivalent obligations will have lower similarity. We evaluate two prompting strategies that differ in how obligations are extracted and aligned.

4.2.1 One-Prompt (1P). This approach performs obligation extraction and alignment in a single step. The model compares both clauses simultaneously and produces concise obligation descriptions. The model assigns these descriptions to one of the four categories. We call them descriptions because they may not reflect the language of either clause, but rather some middle ground to facilitate comparison. To ensure faithfulness, each description must cite the supporting sentence. This method aims to reduce potential misalignment that would be introduced if the obligations were extracted independently. That is, separate obligation extraction may omit useful contextual information when comparing across clauses. We report this method as **1P(model)**.

4.2.2 Two-Prompt (2P). The Two-Prompt approach prompts the model to extract obligations from each clause independently of the other clauses. This creates an interpretable intermediate representation derived from the clause. This is connected to the idea of decomposed prompting and explicit chain-of-thought action [11, 14, 33, 46]. In doing so, the model can focus on identifying the key attributes of each obligation rather than trying to do "everything at

Shared clause prefix	
Further Assurances. Each party shall do and perform, or cause to be done and performed, all such further acts and things, and shall execute and deliver all such other agreements, certificates, instruments and documents, as [any/the] other party may reasonably request in order to carry out the intent and accomplish the purposes of …	
Reference	*… this Agreement and the consummation of the transactions contemplated hereby, including voting for an increase in the Authorized Shares.*
Candidate	*… this Waiver.*
1P(5.2) Output	
Equivalent	*Each party MUST do and perform (or cause to be done and performed) further acts and things, and MUST execute and deliver other agreements/certificates/instruments/documents as the other party may reasonably request, in order to carry out the intent and accomplish the purposes of the governing instrument.*
Reference-only	*(1) Further-assurances duty includes voting for an increase in the Authorized Shares. (2) Further-assurances duty is for carrying out the intent and accomplishing the purposes of this Agreement and consummating the transactions contemplated hereby.*
Candidate-only	*Further-assurances duty is for carrying out the intent and accomplishing the purposes of this Waiver.*
Score	Jaccard = 1/4 = 0.25
L(5.2) Output	
Rank	*6*
Rationale	*(1) Keeps basic further-assurances structure and reasonable-request concept. (2) Purpose is to carry out a 'Waiver' (different instrument and typically narrower context than an agreement with transactions). (3) Omits transaction consummation concept and the specific voting example.*

Table 2: Illustrative example of One-Prompt (1P(5.2)) and Listwise (L(5.2)) outputs for a 'Further Assurances' clause pair (Section 5.2), illustrating structured obligation decomposition vs. Listwise reasoning.

once" as done in the One-Prompt setup. Moreover, these extracted obligations can be reused later for a different comparison without recomputation, as required by the other prompting strategies.

A second prompt then aligns the two obligation sets, labeling the obligations as they fall into one of the four aforementioned categories. The success of this alignment is inherently tied to the quality of the extracted obligations, and mismatches in granularity will propagate into the similarity scores, as we discuss later. We report results as **2P(model|model)**, indicating the models used for extraction and alignment (e.g., **2P(4.1|5.2)**), or as **2P(model)** when the same model is used for both.

4.3 Listwise Similarity (L)

Inspired by work on listwise reranking [1, 24, 44], this approach prompts the model to rank all candidate clauses with respect to the reference clause in a single step. The model is told that clauses may be given the same rank to ensure that it does not make assumptions about intended behavior (i.e., ranks need not be unique). To ground the ranking in the comparison of the reference to the candidate clause, the model must provide a concise justification. In doing so, the goal is to allow the model to reason holistically about relative differences between clauses, as a lawyer might compare multiple clauses simultaneously. We view this approach as an upper bound on effectiveness for the current setup, since it can factor in all available information. Moreover, this approach is close to what we imagine might be employed in a RAG system. Due to issues with positional bias [13, 25, 37], we expect this method to perform worse as more candidate clauses are added, particularly when more realistic data collections (i.e., from hundreds to thousands of contracts) are used. This approach is reported as **L(model)**. To help clarify the distinction between Listwise and Pairwise approaches, Table 2 demonstrates the differences between the two approaches for a held-out clause type used in Section 5.2.

5 Results

Clause	O	G	1P(5.2)	2P(4.1|5.2)	2P(5.2|4.1)	2P(5.2)	L(5.2)
T1	0.56	0.49	0.39	0.56	0.48	0.60	**0.73**
T2	0.40	0.30	0.33	**0.62**	0.46	0.36	0.44
T3	0.40	0.41	0.34	0.49	0.35	0.37	**0.50**
T4	0.75	0.63	0.76	0.75	0.69	0.45	**1.00**
T5	0.56	**0.87**	0.54	0.43	0.66	0.56	0.59
T6	0.45	0.48	**0.73**	0.37	0.46	0.45	0.30
T7	0.43	0.42	**0.64**	0.46	0.31	0.35	0.40
T8	0.54	0.50	0.54	0.55	0.50	**0.71**	0.55
T9	0.52	0.52	0.54	0.48	**0.55**	0.28	0.43
T10	0.57	0.58	0.44	**0.65**	0.51	0.61	0.58

Table 3: RBO for each of the 10 clause types (Table 1 for shorthand mapping). Omitted model configurations are found in the online resource.

Table 3 reports clause-level RBO scores for representative methods.[6] We note that embeddings perform better than anticipated, as they consistently produced reasonable agreement with the lawyer. However, with a single exception discussed later, they are rarely the top-performing method. This suggests that embeddings capture some legal similarity, but are not fully aligned with legal reasoning.

Across clause types, the Listwise method (**L**) has the strongest overall performance. We posit that this is due to the model having all clauses available, which enables reasoning about differences holistically rather than trying to aggregate this through independent pairwise comparisons. In doing so, we see improvements in identifying clause variations. Among prompt-based approaches, *gpt-5.2* provides the largest performance gains, and the Two-Prompt method benefits from using a stronger model for the second prompt. In a subsequent section, we explore the topical variations among methods.

Table 4 reinforces these conclusions as embeddings remain competitive but are generally outperformed by prompt-based approaches, especially Listwise. Notably, both embeddings and Listwise have limited ability to support decision thresholds, whereas Pairwise methods distribute clauses throughout the similarity space, which would be advantageous for workflows that use thresholds for triage.

From Table 5, we see from the bootstrapped 95% confidence intervals (and the standard deviation in Table 4) that there are no significant differences between methods, due in part to the use of only 10 clause types. That said, we note that in a real-world setting, one might know which prompt-based method to use in advance (e.g., due to a predefined workflow). Using this oracle, the average RBO would be 0.679, which is a substantial increase over embeddings (O: 0.52, G: 0.52), but requires trial and error to achieve in practice. Accordingly, while our proposed methods show promise, further refinement is necessary.

Method	RBO	AMI	Rank Diff	Score Diff
O	0.52 (0.11)	0.20 (0.42)	2.72 (2.41)	0.01 (0.01)
G	0.52 (0.16)	0.20 (0.42)	2.77 (2.20)	0.01 (0.01)
1P(5.2)	0.53 (0.15)	0.14 (0.38)	2.40 (2.00)	0.05 (0.10)
2P(4.1|5.2)	0.53 (0.11)	0.06 (0.13)	2.86 (2.26)	0.07 (0.09)
2P(5.2|4.1)	0.50 (0.12)	0.21 (0.37)	2.64 (1.88)	0.08 (0.10)
2P(5.2)	0.47 (0.14)	0.17 (0.34)	3.10 (2.43)	0.07 (0.07)
L(5.2)	0.55 (0.20)	0.35 (0.31)	2.08 (1.90)	–

Table 4: Mean (standard deviation) of RBO, AMI, rank differences, and score differences per method. Listwise produces no pairwise scores (–).

This need for further refinement is exemplified by the cost of performing these calculations (Table 6). Embeddings, as expected,

Method	Lower	Upper
O	0.459	0.582
G	0.439	0.618
1P(5.2)	0.442	0.616
2P(4.1|5.2)	0.470	0.602
2P(5.2|4.1)	0.426	0.567
2P(5.2)	0.394	0.556
L(5.2)	0.446	0.675

Table 5: Bootstrap 95% CIs (10,000 samples) over the observed RBO values. All intervals overlap and no method is significantly different.

[6] Additional results are available in the repository, but were omitted for brevity.

Method	Config	Avg. Input	Avg. Output	Cost / 1k pairs ($)
Embedding	O	279.47	0.00	0.04
One-Prompt	1P(5.2)	1,795.98	439.23	9.30
Listwise†	L(5.2)	3,670.90	1,085.20	2.16
Two-Prompt	*Extract* (4.1)	1,463.80	260.16	5.01
	Extract (5.2)	1,457.80	506.73	9.61
	Match (4.1|5.2)	886.33	76.23	2.60
	Match (5.2|4.1)	1,419.57	179.04	4.30
	Match (5.2)	1,413.57	191.07	5.10

† Ranks all candidates in a single call (100 calls per 1,000 pairs).

Table 6: Average token usage and estimated cost per 1,000 clause pairs by method. For the Two-Prompt approach, extraction is a one-time cost per clause that can be amortized across multiple matching runs; the marginal cost of a new matching pass is therefore the Matching row alone.

are substantially cheaper to generate and use for ranking purposes but lack interpretability. The Listwise approach is substantially cheaper because it performs multiple comparisons at once, whereas the Pairwise approaches incur costs associated with repeatedly identifying and comparing obligations from the reference clause. We note that if this re-ranking process is performed many times, the Two-Prompt approach's obligation extraction may be amortized across those comparisons. In such a case, the cost becomes more reasonable (compared to One-Prompt) but depends on the number of obligations extracted (i.e., with gpt-4.1 generating the fewest). While the cost of prompt-based approaches can be prohibitive in some scenarios, legal tasks are often high-stakes, so the cost may be tolerable, especially with further prompt optimization and compression.

Embeddings provide a stronger baseline than prior work suggested [8], especially when accounting for cost. However, as Section 5.2 shows, embedding-based rankings are not preferred by our lawyer due to the lack of interpretable outputs. Prompt-based methods address this, and further improvements to the prompts may further improve efficiency and effectiveness, which is less easily accomplished for embeddings.

5.1 Clause-Specific Analysis

Previously, we noted that Gemini performed particularly well on T5 (Indemnity) in comparison to the other methods, especially OpenAI. In examining the clauses and the lawyer's notes, we find that the Justia clauses include a second unrelated clause ("Use of Name") in many of the provided clauses. Moreover, many of the clauses are near duplicates of the template clause and each other (i.e., the lawyer identified three sets of clauses, where the most similar are nearly identical to the template). It would appear that the Gemini embeddings are less sensitive to these small differences in the clauses (i.e., the near duplicate clauses have very close cosine similarity values) than OpenAI embeddings. The prompt-based methods, in contrast, appear to be influenced more by the additional clause (which the lawyer ignored in their analysis, as it was not relevant). Interestingly, the Listwise approach shows stronger agreement on the tied ranks (AMI of 0.65), but fails to correctly order the clauses. Given three groups of equivalent clauses and the very minor nuances between them (from the lawyer's notes), we would argue that a human might be reasonably happy with several of these rankings, but we note that this exposes issues with using relatively general prompts.

For T1, One-Prompt struggles with distinctions between entities ("your" vs. "Executive", referring to the same entity in each clause's respective context), but Listwise is able to accommodate these differences and, instead, disagrees with the lawyer's ranking in the importance/materiality of differences (e.g., "assignment at any time" is deemed less material by Listwise). Similarly, for T4, Listwise correctly identifies major differences, but the Pairwise approaches fixate on specific wording differences (e.g., "pay" versus "timely pay", "Tenant" versus "parties"). This reflects a conservative understanding of equivalence and necessitates further refinement.

Listwise is more "opinionated" in its view of drafting and general structure rather than obligation differences for T7, which results in a rigid ordering. While One-Prompt focuses on obligation differences to the reference clause, it yields more ties. It is also worth noting that the lawyer's most legally similar clause also included an unrelated (according to the lawyer) "Acknowledgment" clause, which caused rank variability because methods accounted for it differently. A similar issue arises for T6 where there is a clear discordance between how the lawyer views differences and how the Listwise approach does. For example, in one clause, the lawyer notes that it uses only the term "Employer" and does not specify additional parties, whereas the Listwise approach does not highlight this and instead flags a minor issue with the geographic restriction. This results in a larger rank disparity (rank 6 for lawyer, rank 2 for Listwise) than in the One-Prompt approach (rank 4), which is more focused on identifying and aligning obligations and whose performance results more from nuanced differences in legal language.

Across topics, the performance of Two-Prompt appeared more variable than we had initially expected. Manual inspection of those extractions reveals that, in order to maintain fidelity to clause language, the model often extracts obligations as close to verbatim as possible. When some obligations are expressed with more granularity than others, the resulting asymmetry complicates matching across clauses and artificially lowers similarity (e.g., two extracted obligations agree in parts but not as a whole, such as differing lists of parties). The One-Prompt approach mitigates this issue by identifying and aligning obligations simultaneously, but often uses unique wording to describe the obligation (dissimilar to either clause). But this approach, much like the Listwise approach, may assign different importance (or materiality) to certain aspects that may not align with a lawyer's (e.g., emphasizing certain obligations that a lawyer wouldn't—"MUST do" (model) vs. "shall do" (lawyer), reflected in Table 2 as well).

Variations across clause types were also reflected in the lawyer's notes that they made when ranking clauses. These notes were terse and focused on legally important distinctions (e.g., "*irrevocable consent*," "*assignment at any time*"). These signals were clause-type-specific and sometimes subtle. For example, in T9, "*irrevocable consent*" strongly increased perceived similarity; whereas in T1, adding "*at any time*" reduced it. Our prompts were intentionally general to understand how well these approaches operate and to provide a more balanced comparison to the untuned embeddings. Moreover, some judgments reflect professional interpretation beyond the surface wording (e.g., "*higher [rank] because assignment is permitted if the executive joins an affiliate*"), highlighting the role of domain

expertise in this process. Overall, the variance in the lawyer's notes and the criteria they used to rank clauses indicate that a general-purpose method is unlikely to yield optimal alignment and that methods ought to be adapted to the user and their task.

5.2 Human Validation

To complement quantitative results, we conducted a small qualitative evaluation on an unseen clause type[7] using four methods (**O**, **L(5.2)**, **1P(5.2)**, and **2P(4.1/5.2)**). These results were put into four different spreadsheets, with any additional data (e.g., extracted and matched obligations, Listwise reasons) presented as columns in addition to the rank. Scores were omitted since they are not equivalent across methods. After a short explanation of the methods and their outputs, the lawyer examined the methods and reported their preferences.

Unsurprisingly, the embeddings were least preferred due to the uninterpretability of the ranking and the resulting disagreement with it (i.e., they were unable to determine why rank 3 clauses were ranked differently from rank 6). More surprising was that they viewed the Listwise method as not substantially better, as they struggled to understand rank orders (similar to embeddings), and found the explanations too open to interpretation and not specific enough. The most preferred method was the One-Prompt approach, as it produced a ranking that they agreed with and provided valuable information on how that ranking was calculated (i.e., different categories of obligations). They expressed that some of the wording choices in the obligation descriptions could be improved (" *I would get rid of the capitalized MUST if possible (that gives more importance than warranted to certain sentences).*") but overall saw value in the method. The Two-Prompt approach was a close second but the lawyer found the presentation of the extracted and aligned obligations to be more confusing than the One-Prompt approach. This is not unexpected, as a spreadsheet may not be the most ideal way to present this information. While this is a very small validation experiment, it indicates that the proposed methods have promise and warrant further validation and refinement with additional lawyers.

6 Discussion

Embeddings achieve stronger agreement with the lawyer than anticipated, demonstrating that general-purpose representations capture meaningful aspects of legal similarity, conflicting with prior hypotheses [8]. However, embeddings are not consistently the top-performing approach. Across clause types, at least one prompt-based method yields competitive alignment if not better. Among prompt-based methods, no single approach dominates, though the Listwise method performs particularly well. This likely arises from evaluating all clauses jointly, enabling holistic reasoning about relative differences rather than relying on aligning extracted obligations. However, Listwise reasoning also reveals the model's implicit judgments about legal materiality, which may diverge from those of a lawyer.

Beyond ranking accuracy, prompt-based approaches offer two practical advantages. First, they can distribute clauses more evenly across the similarity space (Table 4), producing more discriminative scores that better support threshold-based triage. Second, obligation-level matching provides transparency into why clauses differ in rank. While these explanations lack the full nuance of legal expertise, they appear to broadly reflect the types of distinctions a lawyer might highlight. This interpretability supports spot-checking and targeted review in ways that embeddings cannot, and was deemed beneficial when our lawyer reviewed the outputs of the different methods. However, prompt-based methods incur a non-trivial cost in token usage, which may be prohibitive for some use cases. On the other hand, legal tasks are often higher stakes than traditional retrieval scenarios and so this cost may be acceptable. Moreover, model costs have declined dramatically over the last several years, and the ability to deploy effective open-source models may also mitigate this high-cost aspect.

Finally, the proposed methods offer a clearer path to customization. Clause-type–specific criteria can be incorporated directly into prompts, enabling alignment with lawyer and task requirements. This would allow practitioners to adapt clause rankings to new tasks and concepts with less manual effort. Achieving comparable specialization with embeddings requires fine-tuning that is likely to require resources (e.g., for data curation) that are outside the realm of possibility for most legal practitioners.

6.1 Limitations and Future Work

Our study does not exhaustively evaluate embedding models, generative models, or prompt designs. Model and prompt selection can materially affect outcomes, and future work should systematically characterize this variability [5, 48]. In particular, clause-type prompt tailoring may better capture legally salient distinctions than general prompt/model optimization. Preliminary follow-on experiments with more granular obligation extraction revealed bimodal behavior driven by inconsistent phrasing, suggesting that normalization strategies, similar to structured information extraction [32], may improve robustness and comparability across clauses.

Additionally, our evaluation relies on a single lawyer. This approach mimics real-life scenarios where a single lawyer (or a small set of authoritative ones) will set precedent for an organization (e.g., standard clause templates and notes encapsulated in a "playbook") that others are expected to follow. However, future studies should include multiple lawyers to quantify variation, understand consensus boundaries, and identify which aspects are lawyer-, clause-type-, and task-specific. In particular, subsequent experiments should use tailored prompts (especially from a particular lawyer's perspective) to more readily encapsulate their conception of relevance, rather than general "one-size-fits-all" prompts as used here. This work intentionally avoided prompt tailoring to maintain a fair comparison with embeddings, which would require additional training data to be fine-tuned.

We note that our use of obligations as the basis for comparison may not encompass all possible interactions between parties in a legal clause. Moreover, combining all possible signals may provide a richer representation than that presented herein. Along these lines, the need to treat named entities as equivalent in the comparison process provides an additional confound from real-world datasets where such entities are more likely to be consistently presented

[7] A set of "Further Assurances" clauses, which state that one or more parties will work together to ensure the agreement is successfully executed.

as they originate from a single company rather than the diverse sources used here.

7 Conclusion

We proposed the idea of a *semantic redline*, a more nuanced counterpart to existing redlines (i.e., edit distance), wherein differences between the legal concepts in clauses are highlighted rather than lexical differences. As a first step towards this goal, we have proposed prompt-based methods to calculate similarity and rank clauses with respect to a reference clause by focusing on the different legal obligations contained therein.

Using lawyer judgments as ground truth, we compared embedding-based and prompt-based methods for ranking legal clauses by similarity to a reference clause. Embeddings provide an effective baseline in the absence of more interpretable solutions. Across clause types, prompt-based methods can produce rankings as good as embeddings, if not better in certain cases. Our results suggest that similarity in legal text is influenced by how legal obligations align and how particular differences are weighted rather than specific semantic interpretations. Approaches that explicitly identify and compare obligations can offer more interpretable rankings than embeddings as validated by the lawyer assessor in our study. Further, prompt-based approaches can be more easily tailored for specific users, tasks, and clause types without needing to retrain models.

References

[1] Mofetoluwa Adeyemi, Akintunde Oladipo, Ronak Pradeep, and Jimmy Lin. 2024. Zero-Shot Cross-Lingual Reranking with Large Language Models for Low-Resource Languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 650–656. doi:10.18653/v1/2024.acl-short.59

[2] Avishek Anand, Procheta Sen, Sourav Saha, Manisha Verma, and Mandar Mitra. 2023. Explainable Information Retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Taipei, Taiwan) *(SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 3448–3451. doi:10.1145/3539618.3594249

[3] Paheli Bhattacharya, Kripabandhu Ghosh, Arindam Pal, and Saptarshi Ghosh. 2022. Legal case document similarity: You need both network and text. *Information Processing & Management* 59, 6 (2022), 103069.

[4] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets straight out of Law School. arXiv:2010.02559 [cs.CL] https://arxiv.org/abs/2010.02559

[5] Gobinda Chowdhury and Sudatta Chowdhury. 2024. AI- and LLM-driven search tools: A paradigm shift in information access for education and research. *Journal of Information Science* 0, 0 (2024), 01655515241284046. doi:10.1177/01655515241284046

[6] Charles L. A. Clarke, Mark D. Smucker, and Alexandra Vtyurina. 2020. Offline Evaluation by Maximum Similarity to an Ideal Ranking. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (Virtual Event, Ireland) *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 225–234. doi:10.1145/3340531.3411915

[7] Matteo Corsi and Julián Urbano. 2024. The Treatment of Ties in Rank-Biased Overlap. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 251–260. doi:10.1145/3626772.3657700

[8] Jonathan Donnelly and Adam Roegiest. 2025. Exploring the Utility of Embedding Similarity for Contract Tasks. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)* (Padua, Italy) *(ICTIR '25)*. Association for Computing Machinery, New York, NY, USA, 401–409. doi:10.1145/3731120.3744609

[9] European Parliament and Council of the European Union. 2022. Regulation (EU) 2022/2554 on Digital Operational Resilience for the Financial Sector (DORA). https://eur-lex.europa.eu/eli/reg/2022/2554/oj. Official Journal of the European Union, L 333, 27 December 2022.

[10] Faegre Drinker Biddle & Reath LLP. 2025. EU Digital Operational Resilience Act: Priorities for 2025. https://www.faegredrinker.com/en/insights/publications/2025/1/eu-digital-operational-resilience-act-priorities-for-2025. Client alert discussing DORA compliance, including contractual risk allocation, liability limits, and indemnities.

[11] Louis Geiger, Danula Hettiachchi, Falk Scholer, and Johanne R. Trippas. 2026. Reasoning with Large Language Models for Relevance Judgements. In *Proceedings of the ACM Conference on International Conference on the Theory of Information Retrieval (ICTIR'26) (ICTIR '26)*. 1–5.

[12] Maura R. Grossman and Gordon V. Cormack. 2011. Inconsistent Responsiveness Determination in Document Review: Difference of Opinion or Human Error? *Pace Law Review* 32, 2 (2011), 267–305. https://digitalcommons.pace.edu

[13] Sebastian Hofstätter, Aldo Lipani, Sophia Althammer, Markus Zlabinger, and Allan Hanbury. 2021. Mitigating the Position Bias of Transformer Models in Passage Re-ranking. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 – April 1, 2021, Proceedings, Part I*. Springer-Verlag, Berlin, Heidelberg, 238–253. doi:10.1007/978-3-030-72113-8_16

[14] Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed Prompting: A Modular Approach for Solving Complex Tasks. arXiv:2210.02406 [cs.CL] https://arxiv.org/abs/2210.02406

[15] Spyretta Leivaditi, Julien Rossi, and Evangelos Kanoulas. 2020. A Benchmark for Lease Contract Review. arXiv:2010.10386 [cs.IR] https://arxiv.org/abs/2010.10386

[16] Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Yueyue Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2023. SAILER: structure-aware pre-trained language model for legal case retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1035–1044.

[17] Haitao Li, Qingyao Ai, Xinyan Han, Jia Chen, Qian Dong, and Yiqun Liu. 2025. DELTA: Pre-Train a Discriminative Encoder for Legal Case Retrieval via Structural Word Alignment. *Proceedings of the AAAI Conference on Artificial Intelligence* 39, 25 (2025), 27072–27080. doi:10.1609/aaai.v39i25.34914

[18] Haitao Li, Yifan Chen, Shuo Miao, Qian Dong, Jia Chen, Yiran Hu, Junjie Chen, Minghao Qin, Yueyue Wu, Yujia Zhou, Qingyao Ai, Yiqun Liu, Cheng Luo, Quan Zhou, Ya Zhang, and Jikun Hu. 2026. LegalOne: A Family of Foundation Models for Reliable Legal Reasoning. arXiv:2602.00642 [cs.CL] https://arxiv.org/abs/2602.00642

[19] Luyao Ma, Yating Zhang, Tianyi Wang, Xiaozhong Liu, Wei Ye, Changlong Sun, and Shikun Zhang. 2021. Legal Judgment Prediction with Multi-Stage Case Representation Learning in the Real Court Setting. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 993–1002. doi:10.1145/3404835.3462945

[20] Dimitris Mamakas, Petros Tsotsi, Ion Androutsopoulos, and Ilias Chalkidis. 2022. Processing Long Legal Documents with Pre-trained Transformers: Modding LegalBERT and Longformer. arXiv:2211.00974 [cs.CL] https://arxiv.org/abs/2211.00974

[21] Afra Nawar, Mohammed Rakib, Salma Abdul Hai, and Sanaulla Haq. 2022. An Open Source Contractual Language Understanding Application Using Machine Learning. In *Proceedings of the First Workshop on Language Technology and Resources for a Fair, Inclusive, and Safe Society within the 13th Language Resources and Evaluation Conference*, Kolawole Adebayo, Rohan Nanda, Kanishk Verma, and Brian Davis (Eds.). European Language Resources Association, Marseille, France, 42–50. https://aclanthology.org/2022.lateraisse-1.6/

[22] Shubham Kumar Nigam, Tanmay Dubey, Noel Shallum, and Arnab Bhattacharya. 2025. Segment First, Retrieve Better: Realistic Legal Search via Rhetorical Role-Based Queries. arXiv:2508.00679 [cs.CL] https://arxiv.org/abs/2508.00679

[23] Andrew Parry, Maik Fröbe, Harrisen Scells, Ferdinand Schlatt, Guglielmo Faggioli, Saber Zerhoudi, Sean MacAvaney, and Eugene Yang. 2025. Variations in Relevance Judgments and the Shelf Life of Test Collections. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Padua, Italy) *(SIGIR '25)*. Association for Computing Machinery, New York, NY, USA, 3387–3397. doi:10.1145/3726302.3730308

[24] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. RankVicuna: Zero-Shot Listwise Document Reranking with Open-Source Large Language Models. arXiv:2309.15088 [cs.IR] https://arxiv.org/abs/2309.15088

[25] Jingfen Qiao, Jin Huang, Xinyu Ma, Shuaiqiang Wang, Dawei Yin, Evangelos Kanoulas, and Andrew Yates. 2026. LLM-Based Listwise Reranking Under the Effect of Positional Bias. In *Advances in Information Retrieval*, Ricardo Campos, Adam Jatowt, Yanyan Lan, Mohammad Aliannejadi, Christine Bauer, Sean MacAvaney, Avishek Anand, Zhaochun Ren, Suzan Verberne, Nan Bai, and Masoud Mansoury (Eds.). Springer Nature Switzerland, Cham, 131–146.

[26] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 1504–1518. doi:10.18653/v1/2024.findings-naacl.97

[27] Jiaming Qu, Jaime Arguello, and Yue Wang. 2020. Towards Explainable Retrieval Models for Precision Medicine Literature Search. In *Proceedings of the*

43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1593–1596. doi:10.1145/3397271.3401277

[28] Jerome Ramos and Carsten Eickhoff. 2020. Search Result Explanations Improve Efficiency and Trust. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1597–1600. doi:10.1145/3397271.3401279

[29] Adam Roegiest, Alexander K. Hudek, and Anne McNulty. 2018. A Dataset and an Examination of Identifying Passages for Due Diligence. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (Ann Arbor, MI, USA) *(SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 465–474. doi:10.1145/3209978.3210015

[30] Adam Roegiest and Edward Lee. 2019. On Tradeoffs Between Document Signature Methods for a Legal Due Diligence Corpus. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 1001–1004. doi:10.1145/3331184.3331311

[31] Adam Roegiest and Anne McNulty. 2019. Variations in Assessor Agreement in Due Diligence. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval* (Glasgow, Scotland UK) *(CHIIR '19)*. Association for Computing Machinery, New York, NY, USA, 243–247. doi:10.1145/3295750.3298945

[32] Sunita Sarawagi. 2008. Information Extraction. *Foundations and Trends in Databases* 1, 3 (2008), 261–377.

[33] Julian A Schnabel, Johanne R Trippas, Falk Scholer, and Danula Hettiachchi. 2025. Multi-stage large language model pipelines can outperform gpt-4o in relevance assessment. In *Companion Proceedings of the ACM on Web Conference 2025*. 1288–1292.

[34] John Stokdyk. 2022. Autonomy: Anatomy of a corporate fraud. https://www.accountingweb.co.uk/business/finance-strategy/autonomy-anatomy-of-a-corporate-fraud

[35] Keet Sugathadasa, Buddhi Ayesha, Nisansa de Silva, Amal Shehan Perera, Vindula Jayawardana, Dimuthu Lakmal, and Madhavi Perera. 2018. Legal Document Retrieval using Document Vector Embeddings and Deep Learning. arXiv:1805.10685 [cs.IR] https://arxiv.org/abs/1805.10685

[36] Zhongxiang Sun, Kepu Zhang, Weijie Yu, Haoyu Wang, and Jun Xu. 2024. Logic Rules as Explanations for Legal Case Retrieval. arXiv:2403.01457 [cs.IR] https://arxiv.org/abs/2403.01457

[37] Raphael Tang, Crystina Zhang, Xueguang Ma, Jimmy Lin, and Ferhan Ture. 2024. Found in the Middle: Permutation Self-Consistency Improves Listwise Ranking in Large Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 2327–2340. doi:10.18653/v1/2024.naacl-long.129

[38] Dietrich Trautmann, Alina Petrova, and Frank Schilder. 2022. Legal prompt engineering for multilingual legal judgement prediction. arXiv:2212.02199

[39] Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2010. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *Journal of Machine Learning Research* 11 (2010), 2837–2854.

[40] Ellen M. Voorhees. 1998. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Melbourne, Australia) *(SIGIR '98)*. Association for Computing Machinery, New York, NY, USA, 315–323. doi:10.1145/290941.291017

[41] Yiquan Wu, Yifei Liu, Weiming Lu, Yating Zhang, Jun Feng, Changlong Sun, Fei Wu, and Kun Kuang. 2022. Towards Interactivity and Interpretability: A Rationale-based Legal Judgment Prediction Framework. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 4787–4799. doi:10.18653/v1/2022.emnlp-main.316

[42] Chaojun Xiao, Xueyu Hu, Zhiyuan Liu, Cunchao Tu, and Maosong Sun. 2021. Lawformer: A pre-trained language model for Chinese legal long documents. *AI Open* 2 (2021), 79–84. doi:10.1016/j.aiopen.2021.06.003

[43] Fangyi Yu, Lee Quartey, and Frank Schilder. 2022. Legal Prompting: Teaching a Language Model to Think Like a Lawyer. arXiv:2212.01326 [cs.CL]

[44] Puxuan Yu, Razieh Rahimi, and James Allan. 2022. Towards Explainable Search Results: A Listwise Explanation Generator. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 669–680. doi:10.1145/3477495.3532067

[45] Weijie Yu, Zhongxiang Sun, Jun Xu, Zhenhua Dong, Xu Chen, Hongteng Xu, and Ji-Rong Wen. 2022. Explainable Legal Case Matching via Inverse Optimal Transport-based Rationale Extraction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 657–668. doi:10.1145/3477495.3531974

[46] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. arXiv:2205.10625 [cs.AI] https://arxiv.org/abs/2205.10625

[47] Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2024. A Setwise Approach for Effective and Highly Efficient Zero-shot Ranking with Large Language Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 38–47. doi:10.1145/3626772.3657813

[48] Jingming Zhuo, Songyang Zhang, Xinyu Fang, Haodong Duan, Dahua Lin, and Kai Chen. 2024. ProSA: Assessing and Understanding the Prompt Sensitivity of LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 1950–1976. doi:10.18653/v1/2024.findings-emnlp.108

A Prompt Templates

This appendix provides the prompt templates used in the experiments in this paper. They are formatted to fit into the paper limits. Not present are the response format instructions that would be appended automatically to these prompts. For the Two-Prompt method, these response formatting instructions also indicate how to align obligations in the matching prompt. Refer to the official repository[8] to view the actual formatting of the prompts.

The following is the Listwise prompt that compares all N clauses to the template/reference clause. As we had no data to refine this prompt, the prompt was tested against the held out clause type (Section 5.2) and was examined by the non-lawyer first author. Accordingly, it was the least prompt-engineered as we had no basis to compare it to.

Listwise

You are a legal clause comparison engine. Your job is to compare a provided template clause, Template, to a list of other clauses provided.

You will rank the additional clauses by their legal similarity to the template clause by providing a rank (with 1 being the most similar). Clauses that are equivalently similar to the template clause may be given the same rank, even if they are different from each other.
In your comparison, you should factor in the parties involved, the obligations, any carve-outs, and any differences in qualification (e.g., "provided that", "from time to time", "except") for these obligations. You should also include a list of reasons for your determination. Each reason should be concise.

Template:
[template]

Clauses to rank:
Clause Name: [clause1 name]
Clause: [clause1 text]

...
Clause Name: [clauseN name]
Clause: [clauseN text]

The following is the One-Prompt prompt and is the most heavily prompt-engineered of the tested methods. The method was heavily tuned against the Donnelly and Roegiest dataset [8] and tried to reflect a structured set of instructions to the model that would aid its reasoning around complex legal issues, but is not tailored to any specific clause. The instructions are legally specific but general-purpose rather than tailored to the assessor or use case. Moreover, preliminary tests of simpler prompts more in line with the other prompts saw diminished effectiveness (on the Donnelly and Roegiest dataset), but not substantially so. However, we include

[8] https://github.com/zuvaai/science/tree/master/ICTIR2026

the tailored prompt to indicate that even legally aware prompts may still benefit from clause- and user-specific tuning. We note that this prompt has parts elided to fit into the paper.

One-Prompt

You are a legal clause comparison engine. Your job is to compare Clause A and Clause B by extracting and aligning their obligations and returning them.

DEFINITIONS
- "Obligation" means any legally operative requirement, prohibition, permission, or commitment. This includes: shall/must/will/agree to, may/is permitted to, shall not/may not/is prohibited from, is entitled to, is responsible for, must ensure, will cause.
- "Legally equivalent obligations" means that, after normalizing defined terms and synonyms, the parties (actor/beneficiary), the deontic modality (must/may/must not), the action, and all material conditions/exceptions are the same in legal effect.
- "Opposite meaning" means the obligations are in direct conflict in legal effect (e.g., permission vs prohibition for same action; required vs prohibited;...), or they reverse a key condition/exception such that the effect flips.
- "Unmatched obligation" means an obligation extracted from one clause has no corresponding aligned obligation in the other clause (neither same nor opposite).

...
INPUTS
You will be given:
- clause_a: a string
- clause_b: a string

INSTRUCTIONS
1) Sentence split each clause. Use the original sentences verbatim for evidence (no rewriting). If a single obligation spans multiple sentences, include all relevant sentences.
2) Extract "obligation atoms" from each clause:...
- Create one atom per distinct duty/permission/prohibition. If a sentence contains multiple duties, split them.
3) Align obligations across Clause A and Clause B:
- Produce pairs when they refer to the same underlying action/subject matter and same actor/beneficiary context.
- Classify each aligned pair as having the same meaning or opposite meaning based on legal effect, not wording.
- If partially overlapping, do NOT force SAME. Prefer:
- SAME only if all material elements match.
- OPPOSITE if the net legal effect conflicts.
- Partial overlaps should be split into discrete obligation atoms to maximize alignment.
- Carve-outs should be treated as separate obligation atom.
4) Unmatched obligations:
- List all atoms from Clause A with no alignment in Clause B, and all atoms from Clause B with no alignment in Clause A.
5) For every result item, include:
- A concise "obligation atom" in neutral legal language (short, not a quotation). - The sentence id of the corresponding clause sentence(s).
6) Quality rules:
- Do not hallucinate parties or conditions not present.
- If parties are implicit (e.g., "each party"), use that phrasing.
- Treat defined terms as-is; do not expand beyond given text.

...
clause_a:
[clause_a text]

clause_b:
[clause_b text]

The remaining two prompts are the obligation extraction and obligation matching prompts used for the Two-Prompt Pairwise method. These prompts were developed after the One-Prompt prompt and reflect a better understanding of the necessary components for a general-purpose matching prompt. The extraction prompt contains few-shot examples generated from the Donnelly and Roegiest dataset [8] as we found that to be more beneficial than explicit rules. The obligation matching prompt is also relatively simple, but we note that the response format instruction (omitted here) provides matching guidance similar to that of One-Prompt but with simpler language. We note that more legal-specific language led to far more granular obligations, causing the matching prompt to fail due to the overwhelming number of obligations that could be generated. Accordingly, we settled on presenting this similar approach rather than other possible variants.

Two-Prompt(Extract)

Your task is to analyze the clause as written, based solely on the operative content, not the heading. List all rights, responsibilities, and obligations based only on what the full text actually states.
The clause may contain a heading, but this heading may not reflect the actual rights, responsibilities, or obligations in the operative language.
Each list item must be semantically exclusive and independent.
Do not infer, interpret, or correct errors. Use the clause exactly as written.
Do not apply legal norms or drafting conventions.
Several examples of clauses and expected obligations follows:
Example 1:
Clause: Exclusivity. During the Term. SIEMENS may develop, manufacture or commercialize ....
Obligations: ["SIEMENS has the right to develop, manufacture, or commercialize a Galectin-3 assay other than the Product during the Term.", ...]
Example 2:
Clause: Assignment. This Agreement may be assigned or otherwise transferred...
Obligations: ["Either Party may assign or transfer the Agreement without the prior written consent of the other Party.", ...]
Example 3:
Clause: RECRUITMENT. During the term of employment and for a period of 12 months...
Obligations: ["The Executive may directly or indirectly hire any of Ceridian's employees who are employed by businesses for which the Executive has or had management responsibility during the term of employment and for a period of 12 months following termination of employment for any reason other than a Change of Control Termination.", ...]

Clause to analyze:
[clause text]

Two-Prompt(Match)

You are a helpful legal assistant that understands legal clauses.
You will be provided two lists of contractual obligations, rights, and responsibilities.
Your task is to align obligations that have exactly the same legal interpretation, those that have completely opposite (i.e., negation), and those that are unmatched from each list.
Ignore differences in entities between the two lists (e.g., assume the company names are the same). Treat all parties between the lists as if they were the same.

The first list of obligations is:
[list1]

The second list of obligations is:
[list2]