

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO - CHILE



“COMPARACIÓN DE RENDIMIENTO DE GRANDES MODELOS DE LENGUAJE (LLMS) EN CLASIFICACIÓN DE DOCUMENTACIÓN DE VALOR REGULATORIO O LEGAL.”

FÉLIX VARGAS CORTÉS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Ricardo Ñanculef
Profesor Correferente: Juan Antonio Abraham Massmann

Julio - 2025



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título; ☐ Tesis de Postgrado;

Título del trabajo: COMPARACIÓN DE RENDIMIENTO DE GRANDES

MODELOS DE LENGUAJE (LLMS) EN CLASIFICACIÓN DE

DOCUMENTACIÓN DE VALOR REGULATORIO O LEGAL

Nombre del candidato(a): Félix Ignacio Vargas Cortés

Carrera / Grado: Ingeniería Civil Informática

Campus: Santiago San Joaquín; **Departamento:** Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Ricardo Ñanculef, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

☒ El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

☐ 6 meses; ☐ 12 meses; ☐ 2 años; ☐ 3 años; ☐ 5 años; ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de

Fecha: 26-08-2025

; Firma:

memoria o tesis:

Estudiante o Candidato(a):

Fecha: 26-08-2025

; Firma:

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

DEDICATORIA

Dedicado a mi mamá.

AGRADECIMIENTOS

Me gustaría agradecer a mi familia, a mi pareja y a todas las amistades que me han acompañado durante este proceso.

RESUMEN

Resumen— Asesorame SpA es una empresa que se encarga de revisar miles de documentos de carácter legal utilizando LLMs, debido a la cantidad de documentos y la importancia de los resultados siempre es importante mantener actualizado tanto las estrategias de *prompts* como los últimos modelos disponibles.

Tras identificar los problemas más importantes para los clientes, se propone la construcción de un benchmark que permita evaluar el desempeño de distintos modelos de LLMs en diversas tareas sobre documentación legal.

El benchmark construido permitió evaluar distintos modelos de manera general y por cada tipo de tarea, permitiendo identificar los mejores modelos para cada tipo de tarea, como también para cada tarea en particular.

Palabras Clave— LLM; Modelos de Lenguaje; Documentos Legales; Benchmarking; Métricas de Evaluación; Procesamiento de Lenguaje Natural

ABSTRACT

Abstract— Asesorame SpA is an enterprise specializing in the review of thousands of legal documents with the aid of LLMs. Given the extensive volume of documents and the paramount importance of the results, maintaining updated prompting strategies and access to the latest available models is consistently crucial.

Upon identifying the most significant challenges for clients, the establishment of a benchmark is proposed to assess the performance of various LLM models across diverse legal documentation tasks.

The developed benchmark enabled the evaluation of different models, both overall and by task type, thereby facilitating the identification of the most suitable models for each task category, as well as for each particular task.

Keywords— LLM; Language Models; Legal Documents; Benchmarking; Evaluation Metrics; Natural Language Processing

GLOSARIO

LLM: Acrónimo de Large Language Model. Su traducción al español es Gran Modelo de Lenguaje.

Benchmark: Del termino en inglés para Pruebas de rendimiento

ÍNDICE DE CONTENIDOS

RESUMEN	iv
ABSTRACT	iv
GLOSARIO	vi
ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABLAS	x
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	2
1.1 Popularización de <i>LLMs</i>	3
1.2 Tipos de operaciones sobre documentos legales	3
1.3 La diferencia con problemas similares	4
1.4 Objetivo general	5
1.5 Objetivos específicos	5
CAPÍTULO 2: MARCO CONCEPTUAL	6
2.1 Definición de Conjuntos Base	7
2.1.1 Elementos básicos	7
2.2 Composición de prompt	9
2.2.1 Rol de Sistema	9
2.2.2 Instrucciones	10
2.2.3 Requisitos	11
2.2.4 Esquema de Respuesta	11
2.2.5 Documento a Revisar	12
2.3 Estado del Arte	13
2.3.1 Clasificación de documentos legales	13
2.3.2 Primeros datasets legales para tareas de revisión	14
2.3.3 Primeros benchmarks legales	16
2.4 Problemas comunes al corregir documentos legales con <i>LLMs</i>	17
2.4.1 LIM (Lost In the Middle)	17
2.4.2 Falsos positivos	18
2.4.3 Comparación de valores esperados	18
2.5 Proceso de refinamiento de <i>prompt</i>	20
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	21
3.1 Requisitos para tareas de revisión	21
3.2 Requisitos para tareas de extracción	26
3.3 Documentos de pruebas	29
3.3.1 Documentos para tareas de revisión	29

3.3.2	Documentos para tareas de extracción	30
3.4	Elaboración de benchmark	30
3.4.1	Construcción de <i>prompts</i>	31
3.4.2	Evaluación de <i>benchmark</i>	32
3.4.3	Acceso a conjunto de pruebas	32
4	VALIDACIÓN DE LA SOLUCIÓN	33
4.1	Análisis de Resultados en Tareas de Extracción	34
4.2	Análisis de Resultados en Tareas de Revisión	36
4.2.1	Rendimiento Balanceado: Métrica F1 (Presente)	36
4.2.2	Mitigación de Riesgo: Métrica de Precisión (Presente)	37
4.2.3	Capacidad de Detección: Métricas de Recall (Presente y Ausente)	38
4.3	Síntesis de métricas	40
	CAPÍTULO 5: CONCLUSIONES	41
5.1	Validación del Marco Propuesto	41
5.2	Resultados y Limitaciones	41
5.3	Trabajo Futuro	42
	REFERENCIAS BIBLIOGRÁFICAS	43

ÍNDICE DE FIGURAS

1	Cantidad de modelos LLM's presentados al público. Fuente: [Naveed <i>et al.</i> , 2023]	3
2	Rol de sistema en prompt para tareas de revisión	9
3	Rol de sistema en prompt para tareas de extracción	9
4	Conjunto de instrucciones para prompt en tareas de revisión	10
5	Conjunto de instrucciones para prompt en tareas de extracción	10
6	Formato de requisitos en prompt	11
7	Esquema de respuesta para tareas de revisión	12
8	Esquema de respuesta para tareas de extracción	12
9	Formato de entrega para documento a revisar en prompt	13
10	Gráficos comparativos precisión contra posición del documento relevante, Fuente:[Liu <i>et al.</i> , 2023]	18
11	Mapa de calor de exactitud en tareas de extracción, Fuente: Elaboración propia.	34
12	Mapa de calor de F1 (Presente) en tareas de revisión, Fuente: Elaboración propia.	36
13	Mapa de calor de Precisión (Presente) en tareas de revisión, Fuente: Elaboración propia.	37
14	Mapa de calor de Recall (Presente) en tareas de revisión, Fuente: Elaboración propia.	38
15	Mapa de calor de Recall (Ausente) en tareas de revisión, Fuente: Elaboración propia.	39
16	Comparación comprehensiva del rendimiento de todos los modelos, Fuente: Elaboración propia.	40

Índice de tablas

1	Distribución base de documentos para operaciones de revisión, Fuente: Elaboración propia.	30
2	Modelos de LLM seleccionados para el <i>benchmark</i> , fuente: Elaboración propia	31

INTRODUCCIÓN

En la era actual de transformación digital, la gestión y análisis de documentación legal representa un desafío significativo para las organizaciones, particularmente en términos de eficiencia operacional y precisión en el procesamiento. Asesorame SpA, una empresa tecnológica chilena, ha desarrollado FAVRAS, una plataforma web especializada en la automatización del análisis y clasificación de documentos legales, dirigida a satisfacer las necesidades de grandes corporaciones que requieren optimizar sus procesos de revisión contractual.

La evolución reciente de los Grandes modelos de lenguaje (Large Language Models o *LLMs*), tal como *ChatGpt*, ha abierto nuevas posibilidades en el procesamiento del lenguaje natural, específicamente en el ámbito legal. Estos modelos, caracterizados por su capacidad de comprensión y generación de texto a nivel casi humano, presentan diferentes niveles de rendimiento según su arquitectura, entrenamiento y aplicación específica. En este contexto, surge la necesidad crítica de establecer metodologías rigurosas para evaluar y comparar el desempeño de distintas *LLMs* en tareas específicas del dominio legal.

La presente investigación se centra en desarrollar un marco comparativo sistemático para evaluar el rendimiento de diversos modelos de *LLMs* en la clasificación, categorización y revisión de documentación legal. Este análisis es fundamental no solo desde una perspectiva académica, sino también desde un punto de vista práctico-empresarial, ya que permite optimizar la implementación de nuevos modelos en sistemas productivos y refinar las estrategias de ingeniería de prompts.

El estudio aborda dos aspectos fundamentales del procesamiento documental: la extracción y validación de información específica. Cada uno de estos aspectos presenta desafíos únicos en términos de precisión y consistencia de procesamiento, factores críticos en el contexto de la revisión contractual corporativa.

La motivación principal de esta investigación surge de la necesidad práctica de Asesorame SpA de mantener un alto estándar de calidad en sus servicios, mientras se adapta rápidamente a los avances en el campo de la inteligencia artificial. La velocidad con la que nuevos modelos de lenguaje son liberados al mercado requiere un método sistemático y eficiente para evaluar su potencial implementación en entornos productivos.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

La empresa Asesorame SpA ha desarrollado FAVRAS, una plataforma especializada en la revisión automatizada de documentos legales mediante el uso de Grandes Modelos de Lenguaje (LLMs). El proceso de revisión requiere un exhaustivo preprocesamiento de los documentos y un riguroso trabajo de Ingeniería de Prompts, lo cual incluye la definición precisa del rol asignado a la LLM, así como la elaboración detallada de definiciones y ejemplos para cada cláusula y/o requisito que deba ser identificado o extraído.

Esto plantea un desafío clave para FAVRAS: ¿cómo medir de manera sistemática y comparable el rendimiento de los prompts diseñados, tanto para tareas de revisión como de extracción, al utilizarlos con nuevos modelos de LLMs que emergen constantemente?

1.1. Popularización de LLMs

En la actualidad, estamos presenciando un aumento significativo[Naveed *et al.*, 2023] en la integración de Grandes Modelos de Lenguaje en una variedad de aplicaciones. Desde la automatización de tareas en la industria legal hasta la generación de contenido en marketing, finanzas[Wu *et al.*, 2023] y la asistencia en la atención al cliente, los LLMs están desempeñando un papel fundamental en la transformación de numerosos sectores. La capacidad de estos modelos para comprender y generar texto de manera precisa y coherente está revolucionando la forma en que interactuamos con la información.

A la fecha existen gran cantidad de LLMs desarrolladas por distintas empresas, entre estas se encuentran *OpenAI*, *Anthropic*, *Google* y *Meta*.

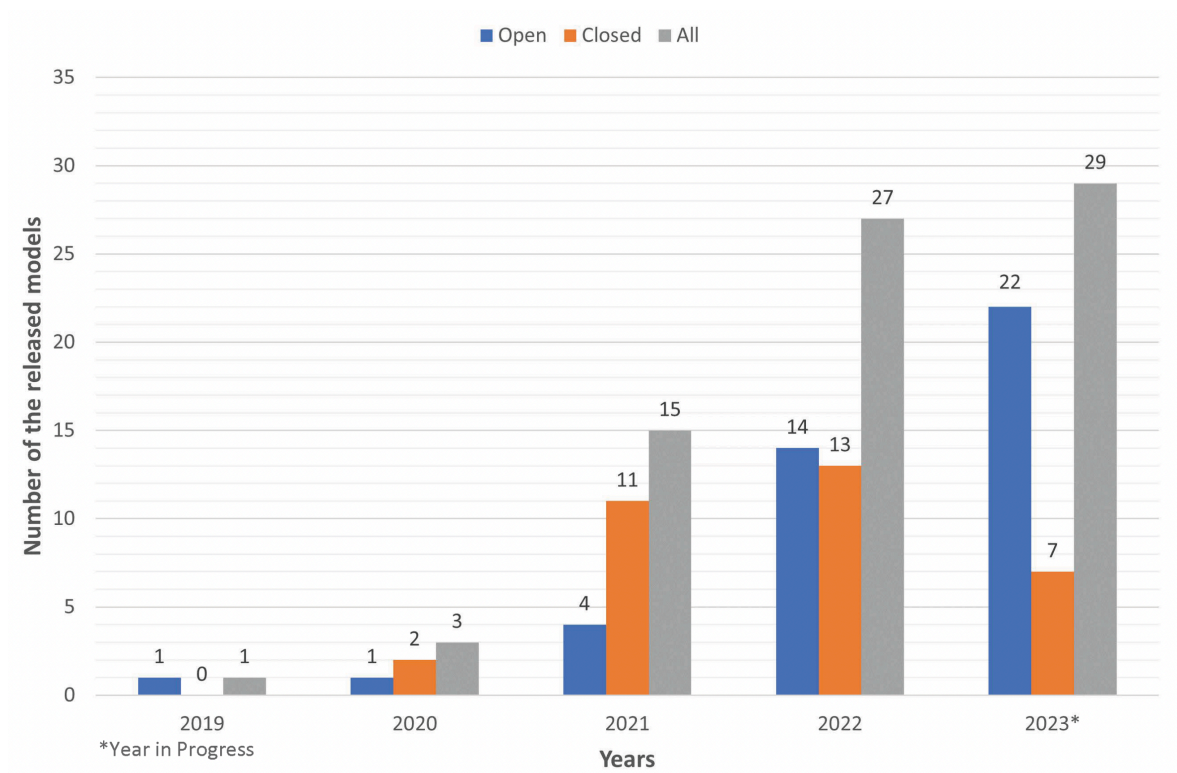


Figura 1: Cantidad de modelos LLM's presentados al público. Fuente: [Naveed *et al.*, 2023]

1.2. Tipos de operaciones sobre documentos legales

En el contexto de FAVRAS, se realizan dos tipos de tareas fundamentales: revisión y extracción. La implementación de cada tipo depende específicamente de las necesidades del cliente:

- **Revisión:** Estas tareas están orientadas a clientes que requieren optimizar sus procesos de revisión de borradores y agilizar la aprobación o firma de contratos con terceros. Estas tareas determinan la presencia, ausencia o estado incompleto de cláusulas o información específica en los documentos.
- **Extracción:** Estas tareas se aplican cuando es necesario construir bases de datos estructuradas a partir de grandes volúmenes de documentos no estructurados (por ejemplo, imágenes digitalizadas de contratos de compraventa de inmuebles). En estos casos, la información relevante puede presentarse en diversos formatos y requiere un nivel significativo de interpretación para su correcta extracción y procesamiento.

Esto correspondería a una descripción a grandes rasgos de las categorías a las que nos enfrentaremos más adelante y que también serán explicadas y ejemplificadas en mayor detalle en el *Marco conceptual*.

1.3. La diferencia con problemas similares

Al utilizar *LLMs* para la revisión de documentos legales, surge el desafío fundamental de evaluar y determinar el desempeño de estos modelos en tareas específicas. Aunque existe una gran cantidad de *benchmarks* para medir el rendimiento de los *LLM*, un porcentaje menor se enfoca en la revisión de documentos legales. De estos, ninguno considera las particularidades de la regulación chilena ni los tipos de documentos específicos que procesa *FAVRAS*.

Al intentar utilizar *benchmarks* del ámbito legal de otros países como EEUU o China (Ver sección 2.8) se presentan dos grandes diferencias/problemas:

- La variabilidad en la presentación de elementos específicos (por ejemplo, cláusulas de confidencialidad en contratos de proveedores B2B).
- La necesidad de comprensión contextual del marco legal y regulatorio del país.

Debido a la falta de un *benchmark* personalizado a las tareas que habitualmente se enfrenta *FAVRAS* se generan pérdidas de tiempo y dinero asociadas a la cantidad de horas que se requieren para probar manualmente distintos documentos y requisitos, guardar las respuestas y finalmente compararlas, todo esto hecho de manera inconsistente y poco reproducibles, ya que, en la mayoría de los casos se utilizan distintos documentos y requisitos para las pruebas manuales, lo que hace que los resultados no sean comparables.

El desarrollo de un *benchmark* para la revisión de documentos legales chilenos, y además de los tipos de documentos a los que se enfrenta *FAVRAS*, requiere de construir cientos de instrucciones y además tener un conjunto de respuestas validadas por expertos en el área para poder comparar los resultados obtenidos.

1.4. Objetivo general

1. Generar *benchmark* de rendimiento para LLMs para cada tarea involucrada en el procesamiento de documentos legales chilenos que realiza FAVRAS.

1.5. Objetivos específicos

1. Definir los tipos de tareas cuyo rendimiento será medido.
2. Construir un conjunto de documentos legales etiquetado con los resultados esperados para cada uno (validados por expertos en el área legal).
3. Seleccionar las métricas de evaluación más relevantes para cada tipo de tarea.
4. Definir la metodología para la construcción de pruebas y comparación de resultados.

CAPÍTULO 2

MARCO CONCEPTUAL

En el contexto de FAVRAS existirán principalmente dos tipos de tareas que se quieren realizar sobre los documentos legales involucrados en el proceso de revisión contractual:

- **Tareas de revisión:** Tareas que se realizan para verificar si un requisito (por lo general una cláusula contractual) está presente en un documento legal, teniendo tres posibles respuestas: *Presente*, *Incompleto* o *Ausente*.
- **Tareas de extracción:** Tareas que se realizan para extraer información específica de un documento legal.

Es importante destacar que en el caso de las tareas de revisión siempre se espera que el modelo encuentre el requisito en el documento, por lo que se espera que la respuesta sea *Presente*. También es importante destacar que los clientes de FAVRAS asumen que en la mayoría de los casos el modelo responderá correctamente, por lo que es importante que se evalúe la capacidad del modelo para no generar falsos positivos, los cuales implicarían que el usuario de FAVRAS(abogado) aprobaría un documento que no cumple con ciertas cláusulas que pueden ser críticas para el cliente y en consecuencia generar problemas legales.

Al revisar documentos legales con *LLMs*, independiente del tipo de tarea que se esté efectuando, existen ciertos procesos que son concomitantes a la tarea en si misma. Para que se puedan desarrollar y ejemplificar estos procesos en mayor detalle es necesario aclarar los siguientes conceptos:

- **LLM (Modelo de Lenguaje de Aprendizaje Automático):** Un modelo de inteligencia artificial diseñado para comprender, interpretar y generar lenguaje humano con una amplia gama de aplicaciones, incluida la revisión de documentos legales.
- **Benchmark (Marco de Evaluación Comparativa):** Conjunto estandarizado de tareas, datos y métricas, diseñado para evaluar y comparar de manera objetiva y reproducible el rendimiento de diferentes sistemas; en este caso, los Grandes Modelos de Lenguaje (LLM).
- **Prompt:** Una entrada de texto proporcionada a un LLM que sirve como disparador o solicitud para que el modelo genere una respuesta o complete una tarea específica.
- **Zero-Shot Prompting:** No proveer al modelo de ningún ejemplo en el prompt para guiarlo en cómo realizar una tarea.
- **Few-Shot Prompting:** Proveer al modelo varios ejemplos (generalmente entre 2 y 5) en el prompt para ilustrar la tarea con más detalle y mejorar su rendimiento.

- **Tokens:** Las unidades básicas de procesamiento en un LLM. Un token puede ser una palabra, parte de una palabra o incluso un carácter, dependiendo del modelo y del enfoque de “tokenización” utilizado.
- **Fine-Tuning:** El proceso de ajustar un modelo pre-entrenado en un conjunto de datos específicos para una tarea en específico mejorando su desempeño en esta.
- **RAG (Retrieval-Augmented-Generation):** Técnica que combina LLMs con sistemas de recuperación de información. Antes de generar una respuesta, el modelo busca datos relevantes en una base de conocimiento externa, mejorando la precisión y relevancia de las respuestas generadas.
- **Base de datos vectorial:** Base de datos que almacena datos como vectores en un espacio multidimensional, representando características numéricas de objetos como texto, imágenes o audio. Permitiendo la búsqueda de elementos utilizando queries de texto que obtienen como resultado el elemento con la similitud semántica más alta.
- **Prompt Engineering:** Se refiere al proceso de diseñar y optimizar los prompts (entradas de texto) para obtener respuestas o comportamientos deseados de una LLM. Esta técnica es crucial para guiar al modelo hacia respuestas más efectivas, precisas y relevantes.
 - *Técnicas:* Incluye la selección de palabras clave, la formulación de preguntas, la estructuración de oraciones y la inclusión de información contextual relevante para guiar al modelo.
 - *Refinamiento de prompts:* Involucra la iteración y experimentación con diferentes formatos y estilos de prompts para mejorar la precisión y relevancia de las respuestas del modelo.

2.1. Definición de Conjuntos Base

A continuación, se definirán las notaciones matemáticas de los elementos fundamentales presentes en cada una de las operaciones que se realizan (en el orden lógico y del proceso), estas se utilizarán a lo largo de la investigación y servirán como medio para generalizar los problemas y métricas que se identifiquen.

2.1.1. Elementos básicos

Se referirá a todo conjunto de documentos legales como D , donde cada documento (un documento corresponderá a todo el texto que contiene un documento legal, por ejemplo: un contrato de proveedores) perteneciente a dicho conjunto d se fragmentará con base en una constante de *tamaño de fragmentos* S , la cual retornará un conjunto de fragmentos F_i :

$D = \{d_1, d_2, \dots, d_n\}$ Conjunto de documentos legales i
 $L(d_i)$: Largo de un documento en cantidad de **tokens**
 S : Constante de tamaño en **tokens** para cada fragmento de un documento
 $\lceil \frac{L(d_i)}{S} \rceil = F_i$: División de documento i que resulta en conjunto de fragmentos
 $F_i = \{f_1^i, f_2^i, \dots, f_n^i\}$: Conjunto de fragmentos para un documento i

Se le llamará **tarea** a cada operación en particular donde se utilizará un requisito (ej: *Encontrar los ruts de los representantes legales*) que se solicitará a un documento o fragmento de este, de esta manera $t_i(r_j, d_k)$ corresponderá a la tarea i de requerir r_j sobre un documento d_k y $t_i(r_j, f_l^k)$ a la tarea i sobre un fragmento f_l de un documento d_k .

$t_i(r_j, d_k)$: Tarea t_i de requerir r_j sobre un documento d_k
 $t_i(r_j, f_l^k)$: Tarea t_i de requerir r_j sobre un fragmento f_l de un documento d_k
 Res_{jk}^i : Respuesta para una tarea t_i de un requisito r_j sobre un documento d_k
 Res_{jkl}^i : Respuesta para una tarea t_i de un requisito r_j sobre un fragmento f_l de un documento d_k

Cada tarea (independiente si se realiza sobre un fragmento o un documento completo) resulta en una respuesta *Res* estructurada que genera la LLM en formato JSON.

Una respuesta para una tarea perteneciente la categoría de *revisión* presenta mayor complejidad que una de extracción, contando con dos elementos, un razonamiento y un resultado, donde el razonamiento correspondería a una explicación del resultado obtenido, y este último pudiendo ser uno de los tres siguientes: **Presente**, **Incompleto** y **Ausente**.

r_j : "Cumplimiento de garantías constitucionales"
 f_l^k : Texto de un fragmento de un contrato marco de proveedores
 $t_i(r_j, f_l^k)$: Tarea t_i de requerir r_j sobre el fragmento f_l de un documento d_k
 Res_{jk}^i : {"Respuesta": "Presente"
"Razonamiento": "Se considera esencial el cumplimiento de las..."}}

Por otra parte, una respuesta para una tarea de extracción corresponde al resultado de lo extraído, por ejemplo:

r_j : "TIPO DE PROPIEDAD"
 d_k : Texto de documento de compraventa del conservador de bienes raíces
 $t_i(r_j, d_k)$: Tarea t_i de requerir r_j sobre el documento d_k
 Res_{jk}^i : "{ "TIPO DE PROPIEDAD": "Departamento" }"

2.2. Composición de prompt

Antes de poder detallar los ciclos de refinamiento de prompts e identificar las métricas para cada categoría de tareas se necesita detallar como se compone un prompt de FAVRAS y el porqué de cada una de sus partes. La nueva estructura del prompt es más simplificada y directa, manteniendo los elementos esenciales para una revisión efectiva de documentos legales (los documentos mostrados en los ejemplos han sido anonimizados por razones de confidencialidad).

2.2.1. Rol de Sistema

Un prompt de FAVRAS cuenta con una estructura simplificada pero efectiva, donde la primera parte corresponde al "Rol de sistema", que cumple la función de contextualizar y dar un objetivo específico al modelo:

Rol de sistema

Eres un abogado chileno especialista en revisión de contratos laborales y de proveedores...

Figura 2: Rol de sistema en prompt para tareas de revisión
Fuente: Elaboración propia.

Rol de sistema

Eres un abogado chileno especialista en revisión de documentos de bienes raíces...

Figura 3: Rol de sistema en prompt para tareas de extracción
Fuente: Elaboración propia.

2.2.2. Instrucciones

El bloque de instrucciones indica a la LLM bajo qué estructura debe responder para cada requisito r y bajo qué criterios/sensibilidad debería considerar algo como *Presente/Incompleto/Ausente* en el caso de las tareas de revisión:

Instrucciones - Revisión

Se te entregará una serie de requisitos que deben buscarse en un contrato, al igual que ejemplos para estos en formato de arreglo. Finalmente, se te proporcionará un esquema de respuesta adecuado. Responde en formato JSON perfectamente parseable, con llaves y valores en castellano, utilizando comillas dobles tanto para las llaves como para sus valores. Para cada requisito deberás responder de la siguiente manera:

```
{
  "nombre_requisito":,
  respuesta: ["Presente" | "Incompleto" | "Ausente"], (enum,str)
  "razonamiento": "Razonamiento del estado del requisito"(str)
}
```

Si los requisitos, por algún motivo, no se ajustan a su descripción, entonces están "Incompletos". Solo si existen muy pocos indicios del requisito, debes responder "Ausente".

Figura 4: Conjunto de instrucciones para prompt en tareas de revisión
Fuente: Elaboración propia.

Instrucciones - Extracción

Se te entregará una serie de requisitos que deben buscarse en un documento, al igual que ejemplos para estos en formato de arreglo, finalmente un schema de respuesta adecuado. Responde en formato JSON perfectamente parseable, con llaves y valores en castellano, utilizando comillas dobles tanto para llaves como sus valores. Para cada requisito deberás responder de la siguiente manera:

```
{
  "nombre_requisito": "valor_respuesta"
}
```

Figura 5: Conjunto de instrucciones para prompt en tareas de extracción
Fuente: Elaboración propia.

2.2.3. Requisitos

Después de las instrucciones, se le entrega al modelo cada requisito r que se necesita en el documento/fragmento, junto con ejemplos que utilizan la técnica de *Few-Shot Prompting* (esta estructura es igual tanto para tareas que involucran revisión como extracción):

Requisitos

Requisitos:

Según Protección de Datos Personales, Proveedor y subcontratados colaborarán para que la contratante cumpla con las leyes aplicables

Ejemplos: [

'El Proveedor se obliga a que tanto él como los terceros que subcontrate colaborarán en todo lo que esté a su alcance para que el Cliente pueda cumplir con las obligaciones legales relativas a la protección de datos personales que le sean aplicables',

'El Proveedor declara que tanto él como sus colaboradores colaborarán en todo lo que esté a su alcance para que el Cliente pueda cumplir las obligaciones legales relativas a la protección de datos personales',

'El Prestador y sus colaboradores colaborarán para que la Contratante pueda cumplir íntegramente con sus obligaciones legales de notificación de brechas de seguridad'

]

Figura 6: Formato de requisitos en prompt

Fuente: Elaboración propia.

2.2.4. Esquema de Respuesta

Antes de agregar al prompt el contenido del documento d o fragmento f pertinente a la tarea, se especifica el schema exacto de cómo debería verse la respuesta en formato JSON, esto es fundamental para garantizar la consistencia y parseabilidad de las respuestas:

Esquema de respuesta - Tareas de revisión

Schema de respuesta:

```
{
  211|Colaboración en cumplimiento de leyes de datos:
    {
      "respuesta":["Presente" | "Incompleto" | "Ausente"],
      "razonamiento":"Razonamiento del estado del requisito"
    }
}
```

Figura 7: Esquema de respuesta para tareas de revisión
Fuente: Elaboración propia.

Esquema de respuesta - Tareas de extracción

Schema de respuesta:

```
{
  231|COMPRADOR: str,
  248|TRANSACCIÓN EN OTRAS MONEDAS: number,
  '229|FECHA: date
}
```

Figura 8: Esquema de respuesta para tareas de extracción
Fuente: Elaboración propia.

Se mantiene el uso de identificadores únicos para cada requisito r (por ejemplo: "232|COMPRADOR"), esto es debido a que en algunos casos la LLM puede distorsionar el nombre del requisito pero no así su identificador, mantener el nombre original del requisito r no es solo importante para poder hacer una evaluación de los resultados sino que también es parte de las necesidades de los clientes de FAVRAS, tener "tags" o r' para ciertos requisitos que no necesariamente son los mismos nombres que funcionan mejor para requerir la información a la LLM, por lo que existe un post-proceso para reemplazar cada r_j por su correspondiente r'_j .

2.2.5. Documento a Revisar

Finalmente, se agrega al prompt el contenido del documento d o fragmento f pertinente al requisito, para este benchmark se han preseleccionado los fragmentos/documentos que

contienen la información relevante para cada requisito, de esta manera se busca que el modelo no tenga que realizar una búsqueda de contenido compleja, sino que pueda responder directamente con la información relevante, evitando así que la *LLM* priorice la información que está al comienzo del documento e ignore la que es de importancia (fenómeno que se explicará mas adelante).

Documento a revisar

Documento a revisar:

5. Protección de Datos Personales

5.1 Las Partes consideran esencial, en la celebración y correcta ejecución del presente Contrato, el cumplimiento de las garantías consagradas en el artículo 19 N°4 de la Constitución Política de la República, y de las disposiciones y principios de la Ley N° 19.628 sobre Protección de la Vida Privada y cualquier otra normativa aplicable o que llegue a ser aplicable durante la vigencia del presente Contrato...

5.2 El Prestador declara y reconoce que la información de los usuarios y/o consumidores, colaboradores, empleados, proveedores y demás a la que tenga acceso la Contratante con ocasión de sus interacciones con aquellos, es parte fundamental de su activo y es de su exclusiva propiedad. En consecuencia, ni el Prestador ni sus trabajadores, representantes o terceros, cualquiera sea la naturaleza de la relación que estos tengan con el Prestador, con los que, o mediante los cuales este último preste los Contrato Definitivo V.B...

Figura 9: Formato de entrega para documento a revisar en prompt
Fuente: Elaboración propia.

2.3. Estado del Arte

2.3.1. Clasificación de documentos legales

El procesamiento de documentos legales mediante *IA* y *Machine Learning* ha ganado relevancia progresivamente, buscando optimizar tareas como la clasificación, extracción de información y análisis predictivo. Sin embargo, la naturaleza intrínseca del lenguaje jurídico, con su terminología específica, ambigüedades y estructuras complejas, presenta desafíos significativos para los modelos computacionales tradicionales.

Un estudio temprano pero ilustrativo de estas dificultades fue presentado por [Nallapati y Manning, 2008] en su trabajo sobre la clasificación de entradas de expedientes judiciales. Los autores demostraron que modelos de *ML* estándar en ese entonces, como las *SVN*, exhibían un rendimiento deficiente en esta tarea específica del dominio legal. Identificaron que estos modelos fallaban en capturar adecuadamente matices semánticos cruciales, lo que denominaron errores “conjuntivos” (error que ocurre cuando el modelo

etiqueta un resultado como “Negativo” al considerar una palabra del texto como negativa, haciendo una similitud con las conjunciones de lógica proposicional) y “disyuntivos”, inherentes a cómo se construyen y entienden las afirmaciones en dichos textos. De manera, se construye un clasificador más simple, basado en lógica proposicional y diseñado explícitamente para abordar estos tipos de errores. Que termina superando a los enfoques de *ML* convencionales.

Las conclusiones del estudio destacan una limitación importante: la necesidad de desarrollar o adaptar modelos con mayor capacidad expresiva para interpretar correctamente las sutilezas y la estructura lógica de los textos legales, un desafío que persiste y motiva la exploración de arquitecturas más avanzadas.

2.3.2. Primeros datasets legales para tareas de revisión

Siguiendo esta línea de investigación sobre los desafíos y las capacidades de modelos de *ML* en el dominio legal, uno de los primeros esfuerzos sistemáticos en la construcción de un dataset y la aplicación de técnicas de “Natural Language Processing” (*NLP*) para predecir resultados judiciales fue el presentado por [Nikolaos Aletras, et al., 2016]. Su trabajo se centró en la tarea de predecir las decisiones del Tribunal Europeo de Derechos Humanos (*TEDH*) basándose únicamente en el contenido textual de los casos.

Para ello se construyó un dataset a partir de las sentencias publicadas por el *TEDH*. Formularon el problema como una tarea de clasificación binaria, donde el objetivo era determinar si, a partir del texto extraído de un caso, se había producido una violación de un artículo específico de la Convención Europea de Derechos Humanos. La información textual se representó mediante secuencias contiguas de palabras (*N-gramas*) y modelado de tópicos. Utilizando estos datos, entrenaron clasificadores, principalmente Máquinas de Vectores de Soporte (*SVM*), para predecir el resultado.

Los resultados de [Nikolaos Aletras, et al., 2016] fueron significativos, demostrando que sus modelos podían predecir las decisiones del *TEDH* con una precisión considerable (alrededor del 79 % en promedio). Un hallazgo clave fue que la sección de “hechos del caso”, tal como era formulada por el propio Tribunal, resultaba ser el factor predictivo más importante, lo que es consistente con teorías del realismo jurídico [Miles y Sunstein, 2007]. Este estudio se considera pionero al ser el primero en abordar sistemáticamente la predicción de fallos en un tribunal internacional importante utilizando exclusivamente el contenido textual de sus decisiones, sentando una base importante para la creación de datasets y el desarrollo de modelos predictivos en el ámbito jurídico.

Avanzando en la creación de recursos para el *NLP* legal, han surgido diversos datasets que abordan tareas y tipos de documentos específicos, reflejando la madurez y especialización del campo. Por ejemplo, [Lippi et al., 2018] presentaron *CLAUDETTE*, un corpus enfocado en el derecho del consumidor que consiste en Términos de Servicio (*ToS*) de plataformas en

línea, etiquetados con el objetivo de detectar automáticamente cláusulas potencialmente abusivas. Este trabajo fue uno de los primeros en automatizar el análisis de este tipo de documentos contractuales. Poco después, introdujeron *LEDGAR*[Tuggener *et al.*, 2020], un corpus de gran magnitud con provisiones legales extraídas de contratos públicos (específicamente, “SEC filings”¹). *LEDGAR* se destaca por su extenso conjunto de más de 12,000 etiquetas, diseñado para la clasificación multietiqueta a gran escala (o extrema) de estas provisiones.

La necesidad de abordar la diversidad lingüística y la complejidad semántica del dominio legal impulsó la creación de datasets aún más especializados. Un año más tarde *Multi-EURLEX*[Chalkidis *et al.*, 2021] es publicado, un notable dataset multilingüe que comprende 65,000 leyes de la Unión Europea, traducidas oficialmente a 23 idiomas y anotadas con etiquetas de la taxonomía *EUROVOC* (Tesauro multilingüe mantenido por la Oficina de Publicaciones de la Unión Europea. Existe en los 24 idiomas oficiales de la Unión Europea). *MultiEURLEX* es fundamental para la investigación en clasificación temática y, crucialmente, para evaluar la transferencia de conocimiento entre diferentes idiomas en un contexto de cero ejemplos (zero-shot cross-lingual transfer).

En el contexto de la comprensión profunda de textos legales, se presenta *CaseHOLD*[Zheng *et al.*, 2021]. Este dataset se enfoca en la tarea fundamental para los juristas de identificar los holdings (fundamentos o decisiones clave) de casos judiciales citados. Compuesto por más de 53,000 preguntas de opción múltiple, *CaseHOLD* fue diseñado para evaluar cómo el pre-entrenamiento específico en el dominio legal beneficia el rendimiento de los modelos de lenguaje en tareas que requieren un razonamiento jurídico detallado.

En el año 2021 se desarrolla el primer dataset (curado por docenas de expertos en el área) para entrenamiento de modelos de *NLP* en tareas de **análisis contractual** *CUAD*[Hendrycks *et al.*, 2021]. Este se enfoca específicamente en el *análisis contractual*, que es el primer paso de la *revisión contractual*, en el cual un abogado y/o asistente revisan cientos de páginas de un contrato y antecedentes para encontrar todas las cláusulas u obligaciones estipuladas. Habiendo recopilado 501 contratos de diversos tipos de los cuáles se han etiquetado 13101 cláusulas, de esta manera se busca que los modelos logren extraer/destacar las cláusulas importantes que encuentren en un contrato.

Dos años más tarde se publica *MAUD*[Wang *et al.*, 2023], dataset etiquetado por expertos para la comprensión lectora de acuerdos de fusión de empresas. Basado en un estudio de la ABA², con +39,000 ejemplos, *MAUD* facilita responder preguntas de opción múltiple sobre cláusulas clave (“deal points”), sirviendo como un *benchmark* para la *NLP* jurídico y la evaluación de modelos en textos contractuales especializados.

¹Cualquier documento financiero que se presente ante la *Comisión de Bolsa y Valores de Estados Unidos* (SEC)

²American Bar Association (ABA): Escuela de abogados de EEUU

2.3.3. Primeros benchmarks legales

LexGLUE [Chalkidis *et al.*, 2022] propone un *benchmark* compuesto por siete datasets existentes, enfocado en estandarizar la evaluación de modelos de *NLP* en diversas tareas de comprensión del lenguaje natural jurídico en inglés, como clasificación de textos y respuesta a preguntas legales. Este trabajo evidenció la necesidad de una evaluación unificada y rigurosa en el dominio.

LEGALBENCH [Guha *et al.*, 2023] es construido colaborativamente por profesionales de derecho, que abarca 162 tareas que cubren seis tipos distintos de razonamiento legal es uno de estos. De forma complementaria, *LexSumm* [Santosh *et al.*, 2024] se enfoca en la evaluación de tareas generativas como el resumen de documentos legales en inglés e introduce *LexT5*, un modelo *seq2seq*³ especializado.

Para abordar la diversidad lingüística, *LEXTREME* [Niklaus *et al.*, 2023] se presenta como un *benchmark* multilingüe (24 idiomas) que, aunque no incluye el español de Chile específicamente, subraya la tendencia hacia la evaluación en múltiples idiomas.

Finalmente, es relevante mencionar *LawBench* [Fei *et al.*, 2023], que aunque centrado en el sistema legal chino, propone una interesante taxonomía de habilidades cognitivas (memorización, comprensión y aplicación) para evaluar *LLMs*, ofreciendo un marco estructurado que podría inspirar el diseño de *benchmarks* para sistemas legales específicos como el chileno.

³Arquitectura de deep learning diseñada para transformar una secuencia de entrada en otra de salida, mediante un codificador y un decodificador

2.4. Problemas comunes al corregir documentos legales con LLMs

Al momento de realizar operaciones sobre documentos legales utilizando LLMs se presentan una variedad de problemas, los cuales a priori pueden parecer simples, pero al intentar solucionarlos surgen una serie de nuevos inconvenientes y sub-problemas relacionados que parecen no tener fin. Estos problemas más generales o transversales (a como se les hará referencia de aquí en adelante) se presentan en el uso de LLMs en general. Se considerará como un error a cualquier respuesta que se desvíe del valor esperado para esta:

$Res_{jk}^{t_i}$: Respuesta esperada al realizar la tarea t_i requiriendo r_j
sobre el documento d_k

$Res_{jkl}^{t_i}$: Respuesta esperada al realizar la tarea t_i requiriendo r_j
sobre el fragmento f_l^k perteneciente al documento d_k

$Res_{jk}^{t_i} \neq Res_{jk}^i$: Error en una tarea sobre un documento

$Res_{jkl}^{t_i} \neq Res_{jkl}^i$: Error en una tarea sobre un fragmento de un documento

2.4.1. LIM (Lost In the Middle)

En primera instancia, sería normal asumir que la manera más sencilla de realizar una tarea t_i involucraría utilizar el documento d_k en su completitud, pero como se ha mencionado antes, este no es el caso, una de las razones de esto es por los costos asociados a utilizar más *tokens* dentro del *prompt*, pero la más importante es que a medida que se incrementa el uso de la ventana de contexto de la LLM (espacio en *tokens* límite para el *prompt*) rápidamente los resultados obtenidos comienzan a decaer en precisión, a esto es a lo que se llama el problema de *Lost in The Middle* [Liu et al., 2023].

A medida que la información relevante para una respuesta precisa está más lejos del comienzo o final de la ventana de contexto, se aprecian disminuciones considerables en la precisión. En el artículo [Liu et al., 2023] se realizan pruebas con distintos modelos a la fecha de publicación, posicionando los fragmentos con información relevante f_l^k en distintas posiciones de la ventana de contexto.

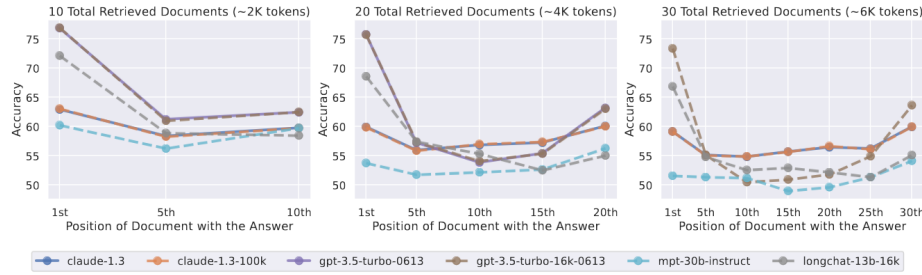


Figura 10: Gráficos comparativos precisión contra posición del documento relevante, Fuente:[Liu *et al.*, 2023]

2.4.2. Falsos positivos

En tareas de revisión, los falsos positivos se producen cuando el modelo responde que un requisito está presente, pero en realidad no lo está. Esto puede ocurrir por varias razones:

- El modelo interpreta mal el contexto del documento.
- El modelo no tiene suficiente contexto para responder correctamente.
- El modelo responde incorrectamente debido a un error en el *prompt*.

En el contexto de revisión de documentos legales, los falsos positivos presentan un problema mayor por sobre los falsos negativos, ya que el usuario de FAVRAS asume que en la mayoría de los casos el modelo responderá correctamente, lo que podría llevar a la aprobación de contratos no conformes, generando un riesgo legal y financiero significativo.

2.4.3. Comparación de valores esperados

En el contexto de la evaluación de modelos de LLMs mediante *benchmarks*, la comparación entre la respuesta generada por el modelo y una respuesta esperada o “ground truth” puede adoptar distintos grados de rigurosidad según el tipo de la tarea evaluada.

Para tareas de revisión, donde se espera que el modelo entregue una respuesta categórica o estructurada (por ejemplo, “Presente”, “Incompleto” o “Ausente”), la evaluación suele realizarse mediante una comparación exacta entre la salida del modelo y la respuesta esperada. En estos casos, la métrica de precisión se calcula considerando únicamente coincidencias 1:1, es decir, la respuesta es considerada correcta solo si coincide exactamente con el valor esperado.

Sin embargo, en tareas de extracción de información, donde el objetivo es recuperar fragmentos textuales específicos o valores dentro de un documento, la comparación exacta puede resultar demasiado restrictiva. Dada la variabilidad inherente al lenguaje natural y la posibilidad de respuestas equivalentes, pero no idénticas en su forma, es común emplear métricas de similitud más flexibles. Entre estas, la distancia de Levenshtein⁴ y otras medidas de similitud textual permiten cuantificar el grado de coincidencia entre la respuesta generada y la esperada, aceptando como correctas aquellas respuestas que, aunque no sean idénticas carácter por carácter, resulten suficientemente próximas según un umbral predefinido. Este enfoque proporciona una evaluación más realista del desempeño del modelo en tareas de extracción.

⁴Métrica que mide la similitud entre dos cadenas de caracteres. Se define como el número mínimo de ediciones de un solo carácter (inserciones, eliminaciones o sustituciones) necesarias para transformar una cadena en otra.

2.5. Proceso de refinamiento de *prompt*

Una de las partes más importantes del proceso iterativo para mejorar las respuestas recibidas es el de *Prompt engineering*, en el caso de las tareas, consiste principalmente en modificar los componentes de un requisito r_j y el *rol de sistema* del *prompt*.

La metodología utilizada para refinar *requisitos* es distinta a la utilizada para los *roles de sistema* o *instrucciones*, estas últimas presentan un menor grado de complejidad.

Cómo práctica general el refinamiento de *roles de sistema* o *instrucciones* es requerido exclusivamente cuando se presentan errores en la estructura de las respuestas, este proceso consiste principalmente en re-escribir con un lenguaje más riguroso la parte que se crea que está fallando.

El refinamiento de *requisitos* es considerablemente más complejo, y varía en cuánto a metodología dependiendo del tipo de error que se presente:

1. Para eliminar posible caso de **LIM**:
 - a) Modificar manualmente documento d_k moviendo el fragmento relevante f_l^k al comienzo del documento o crear un fragmento relevante sintético en caso de que sea un falso positivo.
2. De no ser un problema de **LIM**, modificar f_l^k o d_k de manera que cumpla con el resultado esperado (en caso de que no cumplan con lo requerido) y probar nuevamente.
3. Modificar r_j agregando símiles de la $Res_{j_k}^i$ como ejemplos y/o modificando la descripción de r_j agregando más detalles a esta. Luego de esto, repetir nuevamente el paso 2.
4. Cambiar modelo de *LLM* y repetir proceso.
5. En el caso de no poder lograr la respuesta esperada se procede a restituir el requisito.

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

Para resolver el problema de no poder medir el rendimiento de nuevos modelos de *LLM* en tareas de revisión y extracción de documentos legales Chilenos de manera consistente y rápida, se construirá un *benchmark* que permitirá evaluar el rendimiento de manera adecuada, permitiendo así a FAVRAS conocer rápidamente el rendimiento de nuevos modelos de *LLM*, específicamente en los *requisitos* que son de interés (en vez de basarse en *benchmarks* con un enfoque más general), y de esta forma poder decidir si integrarlos a la plataforma con certeza de que cumplen con los requerimientos mínimos de “precisión” que son prometidos contractualmente.

Para la construcción del *benchmark* primero se debe seleccionar que *requisitos* son las más relevantes para determinar si un modelo es adecuado, además de esto, y más importante aún, es que los documentos sobre los cuales se evaluarán los requisitos deben haber sido previamente analizados y etiquetados por expertos en el área, por lo cual no solo la cantidad de documentos que se pueden utilizar para el *benchmark* es limitada, sino que también la cantidad de *requisitos*.

Primero comenzaremos por definir que *requisitos* se evaluarán para cada tipo de tarea, cada uno de estos cuenta con un identificador (*ID*) único que permitirá identificarlos.

3.1. Requisitos para tareas de revisión

Los siguientes requisitos han sido etiquetados por expertos en el área para la evaluación de documentos legales correspondientes a *Contratos de Proveedores* (Acuerdos legales que por lo general son entre una empresa y un proveedor de bienes o servicios). Cada requisito se describe en términos de su condición de presencia; se detalla qué cláusulas o estipulaciones deben encontrarse en el texto para que el requisito se considere **Presente** en el resultado de la tarea. La ausencia de una cláusula explícita o una mención parcial o ambigua llevará a una calificación **Ausente** o **Incompleta**, respectivamente.

1. Eliminación o devolución de datos al término del contrato (ID: 213)

Descripción: Este requisito se considera **Presente** si el contrato estipula explícitamente la obligación del proveedor (la contratada) de eliminar, suprimir, destruir o devolver los datos e información pertenecientes al cliente (la contratante) una vez finalizada la vigencia del contrato.

Ejemplos de cláusulas que satisfacen este requisito:

- “Terminado el presente Contrato, el Proveedor suprimirá y/o inutilizará de sus bases de datos, todo dato personal transmitido por el Cliente.”
- “El Proveedor deberá devolver a la Administradora o destruir, si así se requiere, la Información Confidencial una vez terminado el presente Contrato.”

2. Eliminación de datos: aplicación a terceros subcontratados (ID: 214)

Descripción: Se evalúa como **Presente** si el contrato extiende la obligación de eliminación o devolución de datos a cualquier tercero que el proveedor haya subcontratado para la prestación de los servicios.

Ejemplos de cláusulas que satisfacen este requisito:

- “El Proveedor deberá, a la terminación del contrato, suprimir de sus bases de datos, y hacer que supriman de las bases de datos de los terceros que haya subcontratado, todo dato personal al que haya tenido acceso...”
- “Terminado el presente Contrato, el Proveedor y los terceros que haya subcontratado suprimirán y/o inutilizarán de sus bases de datos todo dato personal transmitido por el Cliente...”

3. Mantenimiento de la confidencialidad post-contratual (ID: 217)

Descripción: El requisito se considera **Presente** cuando el documento legal establece que la obligación de mantener la confidencialidad de la información persiste más allá de la terminación del contrato, ya sea de forma indefinida o por un período de tiempo específico.

Ejemplos de cláusulas que satisfacen este requisito:

- “Sin perjuicio de la terminación del presente Contrato, por cualquier causa, la obligación de confidencialidad se mantendrá vigente o subsistirá de forma indefinida.”
- “Las obligaciones de confidencialidad pactadas en este Contrato tendrán vigencia durante la vigencia del Contrato y por el período de 5 años.”

4. Nivel mínimo de diligencia (ID: 218)

Descripción: Este requisito se encuentra **Presente** si el contrato especifica un estándar de cuidado o nivel de diligencia con el que el proveedor y sus subcontratados deben resguardar la confidencialidad de la información.

Ejemplos de cláusulas que satisfacen este requisito:

- “Con la diligencia de un buen padre de familia.”
- “Con el máximo nivel de diligencia.”

5. Mención de la ley de protección de datos personales (ID: 220)

Descripción: Se considera **Presente** si el contrato hace referencia explícita a la obligación de cumplir con la normativa de protección de datos personales vigente en Chile (como la Ley N°19.628) u otras normativas aplicables.

Ejemplos de cláusulas que satisfacen este requisito:

- “Se considera esencial el cumplimiento de las disposiciones y principios de la ley sobre la Protección de la Vida Privada.”
- “Declaramos cumplir las disposiciones y principios de la Ley N°19628 sobre Protección de la Vida Privada.”

6. Adopción de medidas de seguridad y confidencialidad (ID: 210)

Descripción: El requisito está **Presente** si el contrato estipula que el proveedor debe adoptar y mantener medidas de seguridad, técnicas u organizativas, para proteger los datos, resguardar la confidencialidad y prevenir accesos no autorizados, fugas o pérdidas de información.

Ejemplos de cláusulas que satisfacen este requisito:

- “El Proveedor se obliga a la adopción y mantenimiento de las medidas de seguridad, confidencialidad y protección necesarias para garantizar la protección de los datos personales transmitidos por el Cliente.”
- “Cada Parte se compromete a mantener la Información Confidencial de la otra parte en forma estrictamente confidencial y a adoptar recaudos razonables para proteger dicha Información Confidencial.”

7. Deber de colaboración en cumplimiento normativo (ID: 211)

Descripción: Se considera **Presente** si el texto contractual indica que el proveedor (y sus subcontratados, si aplica) colaborará con el cliente para que este último pueda cumplir con las leyes y obligaciones que le son aplicables en materia de protección de datos.

Ejemplos de cláusulas que satisfacen este requisito:

- “El Proveedor se obliga a que tanto él como los terceros que subcontrate colaborarán en todo lo que esté a su alcance para que el Cliente pueda cumplir con las obligaciones legales relativas a la protección de datos personales que le sean aplicables.”

8. Responsabilidad del proveedor por incumplimiento de terceros (ID: 212)

Descripción: El requisito está **Presente** si el contrato establece explícitamente que el proveedor será responsable ante el cliente por cualquier incumplimiento de las obligaciones de protección de datos o confidencialidad en que incurran los terceros que haya subcontratado.

Ejemplos de cláusulas que satisfacen este requisito:

- “El Proveedor será responsable frente al Cliente de cualquier incumplimiento por parte de los terceros que subcontrate.”

9. Mención de garantías constitucionales (ID: 219)

Descripción: Se evalúa como **Presente** si el contrato hace alusión explícita al cumplimiento de las garantías constitucionales sobre protección de la vida privada, usualmente refiriéndose al artículo 19 N°4 de la Constitución Política de la República de Chile.

Ejemplos de cláusulas que satisfacen este requisito:

- “Se considera esencial el cumplimiento de las garantías consagradas en el artículo 19 N°4 de la Constitución Política de la República de Chile.”

10. Deber de indemnización por uso indebido de datos (ID: 215)

Descripción: Este requisito se considera **Presente** si el contrato establece la obligación del proveedor de indemnizar al cliente por cualquier perjuicio, daño, reclamo o sanción derivado de un incumplimiento de sus responsabilidades en materia de protección de datos o confidencialidad.

Ejemplos de cláusulas que satisfacen este requisito:

- “El Prestador se obliga a indemnizar y a mantener indemne a la Contratante [...] con respecto a cualquier reclamación, queja, demanda, acción o, pretensión, por cualquier perjuicio, daño, pasivo y/o contingencia.”

11. Uso exclusivo de la información para fines del contrato (ID: 224)

Descripción: El requisito está **Presente** si se indica de manera explícita que la información y los datos del cliente solo pueden ser utilizados por el proveedor para los fines estrictamente necesarios para la ejecución de los servicios objeto del contrato.

Ejemplos de cláusulas que satisfacen este requisito:

- “El Proveedor podrá solamente acceder y utilizar la información de Fabelab, en cuanto dicho uso resulte estrictamente indispensable para la prestación de los servicios contratados.”

12. Reconocimiento explícito de la confidencialidad (ID: 223)

Descripción: Se considera **Presente** si el contrato contiene una declaración donde el proveedor reconoce el carácter confidencial de la información del cliente y se compromete a aplicar los resguardos necesarios para asegurar su integridad y disponibilidad.

Ejemplos de cláusulas que satisfacen este requisito:

- “Toda información relacionada con los clientes de las empresas y sociedades del Grupo Falabella será tratada como información confidencial.”

13. Restricción de difusión de la información (ID: 226)

Descripción: El requisito está **Presente** si se prohíbe explícitamente al proveedor, su personal y sus subcontratados difundir, comunicar o utilizar la información del cliente para fines distintos a los estipulados en el contrato.

Ejemplos de cláusulas que satisfacen este requisito:

- “Ni el Proveedor ni sus trabajadores, representantes o terceros, podrán bajo ningún medio o circunstancia, difundir, informar o utilizar dicha información, para cualquier otro fin distinto a los requeridos para la prestación de los servicios objeto de este Contrato.”

14. Consecuencias por violación de confidencialidad (ID: 227)

Descripción: Se evalúa como **Presente** si el contrato define consecuencias específicas para el proveedor en caso de incumplimiento de las obligaciones de seguridad, confidencialidad o protección de datos. Estas consecuencias pueden incluir la terminación del contrato, la aplicación de multas o el inicio de acciones legales.

Ejemplos de cláusulas que satisfacen este requisito:

- “El incumplimiento de cualquiera de los compromisos [...] será motivo suficiente para la terminación unilateral del Contrato por parte del Cliente, sin necesidad de sentencia judicial...”
- “Cualquier infracción o contravención a lo señalado en la cláusula de Confidencialidad, dará derecho a Falabella a iniciar las acciones legales que procedan...”

15. Declaración sobre la proveniencia lícita de los datos (ID: 390)

Descripción: El requisito está **Presente** si el proveedor declara en el contrato que los datos que utiliza o transfiere han sido obtenidos de manera lícita, en conformidad con la ley, o que posee las licencias válidas para su uso (aplicable a software o propiedad intelectual).

Ejemplos de cláusulas que satisfacen este requisito:

- “El proveedor declara que puede comunicar y transferir lícitamente todas las bases de datos objeto del Contrato y que han sido obtenidas con observancia de la ley.”
- “La Empresa se obliga a usar única y exclusivamente sistemas informáticos de su propiedad o válidamente licenciados.”

3.2. Requisitos para tareas de extracción

Los siguientes requisitos han sido etiquetados por expertos en el área para la evaluación de documentos legales provenientes del Conservador de Bienes Raíces (CBR), específicamente *escrituras de compraventa* (escritura pública que formaliza la transferencia de la propiedad de un bien, generalmente un inmueble, de un vendedor a un comprador). Cada requisito se describe en términos de su semblanza y formato, por lo que se espera que el modelo sea capaz de extraer la información de manera correcta y en el formato esperado, esto es, a diferencia de las tareas de revisión donde existe incertidumbre sobre la presencia de un requisito, en este caso el interés del cliente es el contenido del documento y no el cumplimiento legal de este.

1. Naturaleza de la Transacción (ID: 230)

Descripción: Extraer el tipo de acto o contrato jurídico que da origen a la inscripción en el Conservador de Bienes Raíces. El resultado debe ser una de las categorías predefinidas.

Valores esperados: COMPRAVENTA, FUSIÓN, HERENCIA, POSESIÓN EFECTIVA, etc.

2. Nombre del Comprador (ID: 231)

Descripción: Extraer el nombre completo de la persona natural o la razón social de la persona jurídica que adquiere el dominio del inmueble.

Ejemplos de extracción: ABEL GONZALEZ NIETO, SOCIEDAD INMOBILIARIA E INVERSIONES X.

3. RUT del Comprador (ID: 232)

Descripción: Extraer el Rol Único Tributario (RUT) de la parte compradora. El valor debe ser normalizado al formato XX.XXX.XXX-X. Se deben ignorar identificadores de extranjeros si el texto lo especifica.

Ejemplos de normalización:

- “Rol Único Tributario Número noventa y nueve millones quinientos mil cuatrocientos diez guion cero” → 99.500.410-0.

4. Nombre del Vendedor (ID: 244)

Descripción: Extraer el nombre completo de la persona natural o la razón social de la persona jurídica que transfiere el dominio del inmueble.

5. RUT del Vendedor (ID: 245)

Descripción: Extraer el Rol Único Tributario (RUT) de la parte vendedora, normalizado al formato XX.XXX.XXX-X.

6. Nombre del Requiriente (ID: 254)

Descripción: Extraer el nombre completo o razón social de la persona o entidad que solicita la inscripción de la transacción.

7. Fecha de la Escritura (ID: 229)

Descripción: Extraer la fecha de la escritura pública que sirve como título para la transacción (no la fecha de inscripción en el Conservador). El formato de salida debe ser *dd-mm-aaaa*. Se requiere la conversión de fechas escritas en palabras.

Ejemplo de normalización: "Según escritura de fecha cinco de junio del año dos mil veintidós" → 05-06-2022.

8. Fecha de Registro (ID: 4949)

Descripción: Extraer la fecha en que se realiza el registro en el Conservador, que suele ser la primera fecha mencionada en el documento. El formato de salida debe ser *dd-mm-aaaa*.

Ejemplo de normalización: "Santiago, siete de mayo del año dos mil veinticuatro" → 07-05-2024.

9. Tipo de Propiedad (ID: 234)

Descripción: Clasificar la propiedad transaccionada en una o más de las categorías predefinidas (ej. *DEPARTAMENTO*, *ESTACIONAMIENTO*, *BODEGA*). Si un inmueble incluye múltiples tipos, estos deben ser concatenados.

Ejemplos de clasificación: *DEPARTAMENTO*, *DEPARTAMENTO + BODEGA + ESTACIONAMIENTO*.

10. Dirección del Inmueble (ID: 235)

Descripción: Extraer la calle y numeración principal de la propiedad. Se debe normalizar el nombre de la calle y el número a un formato estándar.

Ejemplo de extracción: *SIMON BOLIVAR #2641*.

11. Identificación Específica de la Propiedad (ID: 236)

Descripción: Extraer los identificadores únicos de las unidades inmobiliarias (departamento, bodega, estacionamiento, oficina), incluyendo su numeración y/o letra. Este requisito busca aislar el identificador específico de la unidad, excluyendo la calle.

Ejemplo de normalización: "Departamento número veintitrés del edificio colectivo número veintitrés" → *Departamento N°23 Edificio colectivo N°23*.

12. Comuna (ID: 237)

Descripción: Extraer el nombre de la comuna donde se ubica la propiedad.

13. Rol de Avalúo Principal (ID: 249)

Descripción: Extraer el Rol de Avalúo Fiscal del inmueble principal, normalizándolo al formato numérico con guión.

Ejemplo de normalización: "Rol de avalúo número dos mil uno guión quinientos sesenta y nueve" → 2001-569.

14. Monto de la Transacción en Pesos Chilenos (ID: 247)

Descripción: Extraer el monto de la transacción si está expresado en pesos chilenos. El valor debe incluir el símbolo de la moneda y los separadores de miles.

15. Monto de la Transacción en Unidades de Fomento (UF) (ID: 246)

Descripción: Extraer y normalizar el monto de la transacción si está expresado en Unidades de Fomento (UF). El valor extraído debe ser únicamente el número (entero o decimal), convirtiendo montos en palabras a su representación numérica.

Ejemplo de normalización: "TRESCIENTOS CUARENTA COMA SESENTA Y CINCO UNIDADES DE FOMENTO" → 340,75.

16. Monto de la Transacción en Otras Monedas (ID: 248)

Descripción: Extraer el monto y la unidad de cuenta si la transacción se realiza en una moneda o unidad distinta de CLP o UF.

Ejemplos de extracción: USD 5.000, 16.443 CUOTAS DE AHORRO.

17. Nombre del Proyecto Inmobiliario (ID: 242)

Descripción: Extraer el nombre del conjunto habitacional, edificio o condominio al que pertenece la propiedad, si se menciona explícitamente.

18. Número de Propiedades (ID: 10021)

Descripción: Realizar un cómputo de la cantidad total de unidades inmobiliarias individuales (departamentos, bodegas, estacionamientos, etc.) que se transfieren en el acto.

Ejemplo de cómputo: "Departamento N° 200 Bodega N° 213 Estacionamiento N°203" → 3.

19. Identificadores de Lote, Manzana y Sitio (IDs: 238, 239, 240)

Descripción: Extraer los identificadores de Lote, Manzana y/o Sitio asociados a la propiedad. Una regla crítica es ignorar las menciones de estos elementos cuando aparecen en la descripción de los deslindes (propiedades colindantes).

Ejemplo de extracción (Lote): “Que corresponde al Lote A uno de la manzana H” → A-1.

3.3. Documentos de pruebas

Para la construcción del *benchmark* se utilizarán documentos legales que han sido previamente analizados y etiquetados por expertos en el área, de los cuales se cuenta con 344 documentos para tareas de revisión y 1500 documentos para tareas de extracción.

3.3.1. Documentos para tareas de revisión

Los documentos para tareas de revisión consisten exclusivamente a *contratos de proveedores*. Existe una gran variabilidad entre los documentos dentro de lo posible siendo todos estos pertenecientes a la misma categoría de documento (contratos de proveedores), lo cual pone a prueba la capacidad de abstracción de los modelos para generalizar los requisitos.

De los 344 documentos y 15 tareas que componen el *benchmark* de operaciones de *revisión* se obtienen 5160 comparaciones, de las cuales se detallarán por tarea la distribución de etiquetas:

ID	Presente	Ausente	Incompleto
210	115	129	100
211	121	208	15
212	201	124	19
213	188	104	6
214	128	170	46
215	38	269	37
217	221	102	21
218	168	151	25
219	144	178	22
220	159	135	50
223	202	73	69
224	219	113	12
226	198	97	49
227	165	143	36
390	97	244	3
Total	2364	2240	510

Tabla 1: Distribución base de documentos para operaciones de revisión, Fuente: Elaboración propia.

3.3.2. Documentos para tareas de extracción

Como fue mencionado anteriormente todos los documentos para tareas de extracción corresponden a un único tipo de documento (*escritura de compraventa*) proveniente del CBR, esto por un lado se debe a las necesidades de los clientes de FAVRAS, y por otro lado porque corresponde al único tipo de documento proveniente del CBR que permite para cada tarea de extracción obtener respuestas no nulas (con un valor distinto a “Ausente/No disponible”, etc.).

La estructura de estos documentos en cuanto a su contenido es idéntica entre sí, esto se debe al uso de plantillas para la generación de estos, por lo que puede esperarse que los modelos tengan mayoritariamente problemas en cuanto al formato de las respuestas más que en identificarlas correctamente.

3.4. Elaboración de benchmark

Para la elaboración de las pruebas se seleccionará un número determinado de modelos de LLM, que son los siguientes:

Estos modelos han sido seleccionados principalmente por ser los más recientes, con mejores

OpenAI	Gemini
GPT-4.1	flash-2.0
GPT-4.1-mini	pro-2.5
GPT-4.1-nano	flash-2.5
GPT-4o	
GPT-4o-mini	
o3-mini	
o4-mini	

Tabla 2: Modelos de LLM seleccionados para el *benchmark*, fuente: Elaboración propia

resultados en *benchmarks* generales y costos, además de contar con la capacidad de recibir trabajos masivos (batch), lo cual además de ser un facilitador para realizar pruebas también es un requisito para algunos clientes de FAVRAS.

Adicionalmente como requisito para considerar los modelos se toma en cuenta la disposición de créditos para *startup* que los proveedores de los modelos hayan ofrecido a FAVRAS.

3.4.1. Construcción de *prompts*

Para evitar problemas de *LIM* cada *prompt* se construirá considerando solamente los fragmentos relevantes de cada documento para cada requisito, haciendo previamente un proceso de fragmentación y vectorización de los documentos para luego realizar una búsqueda semántica de los fragmentos relevantes utilizando los requisitos en conjunto a sus descripciones y ejemplos. Una vez los documentos han sido fragmentados y vinculados al requisito correspondiente se construye el *prompt* acorde a los componentes mencionados anteriormente.

Algo importante a considerar en la construcción de cada *prompt*, en particular en el proceso de obtención de fragmentos relevantes es que se extraerán siempre los primeros dos fragmentos más relevantes del documento al realizar la búsqueda semántica, donde cada fragmento tendrá un largo *S* de 5500 *tokens* (16000 caracteres aproximadamente), para cada fragmento se considerará un *solapamiento* u *overlap* de 50 *tokens* (siendo *overlap* el número de *tokens* que se superponen entre cada fragmento) permitiendo así mantener la coherencia de los fragmentos.

Finalmente, cabe destacar que solo para los *prompts* relacionados a tareas de revisión serán compuestos con fragmentos de los documentos, mientras que para las tareas de extracción se utilizará el documento completo. Esto se debe a que los documentos para tareas de extracción que se utilizarán en este *benchmark* cuentan con una extensión lo suficientemente breve como para que se eviten problemas de *LIM*.

3.4.2. Evaluación de *benchmark*

Para este proceso se utilizará un *script* para crear cada archivo para revisiones *batch* en formato **JSON Lines**⁵ (formato solicitado para revisiones masivas de cada proveedor), finalmente los resultados de cada trabajo *batch* se compararán con los resultados esperados para el requisito y documento correspondiente.

De estas comparaciones se medirán las siguientes métricas que se consideran relevantes para tareas de revisión:

- **Precisión Presentes (P_p):** Proporción de predicciones positivas que fueron correctas. Se calcula como $VP/(VP + FP)$, donde VP son verdaderos positivos y FP son falsos positivos. Una precisión alta indica baja tasa de falsos positivos.
- **Recall Presentes (R_p):** Proporción de casos positivos reales que fueron correctamente identificados. Se calcula como $VP/(VP + FN)$, donde FN son falsos negativos.
- **F1 Presentes ($F1_p$):** Media armónica entre precisión y recall. Proporciona una métrica balanceada especialmente útil cuando existe un “trade-off” entre *Precisión* y *Recall*.
- **Recall Ausente (R_a):** Recall de clase “Ausente”, que corresponde a la cantidad de predicciones correctas de clase “Ausente” sobre el total de casos verdaderamente etiquetados como “Ausente”, permitiendo medir la capacidad del modelo para identificar correctamente la ausencia de elementos requeridos.

Por otro lado, para tareas de extracción la única métrica que se considerará es la exactitud:

- **Exactitud (E):** Cantidad de predicciones correctas sobre el total de predicciones, en las extracciones se considerará como correcta una respuesta para una tarea con respecto a la respuesta esperada si, y solo si es exactamente igual a la respuesta esperada o presenta una similitud de al menos un 80 % utilizando distancia coseno entre ambos vectores.

3.4.3. Acceso a conjunto de pruebas

Debido a la confidencialidad de los documentos y sus contenidos no es posible la publicación de los conjuntos de pruebas que se utilizarán hasta que estos hayan sido anonimizados y posteriormente autorizados por los clientes de FAVRAS para su liberación.

⁵Formato similar a JSON donde cada línea corresponde a un objeto JSON válido

VALIDACIÓN DE LA SOLUCIÓN

En este capítulo se presentan y analizan los resultados obtenidos tras ejecutar el *benchmark* desarrollado sobre el conjunto de modelos *LLM* seleccionados, documentos y tareas definidas en el capítulo 3.1 y 3.2. El propósito es validar la eficiencia del *benchmark* para diferenciar el rendimiento de los modelos en las tareas de extracción y revisión de documentos legales chilenos.

Los resultados para cada métrica se presentarán utilizando mapas de calor. Donde el eje **X** representa los nombres de cada *LLM* y el eje **Y** los nombres de cada tarea (en conjunto a su identificador).

4.1. Análisis de Resultados en Tareas de Extracción

La primera evaluación se centra en las tareas de extracción, cuya métrica principal es la Exactitud.

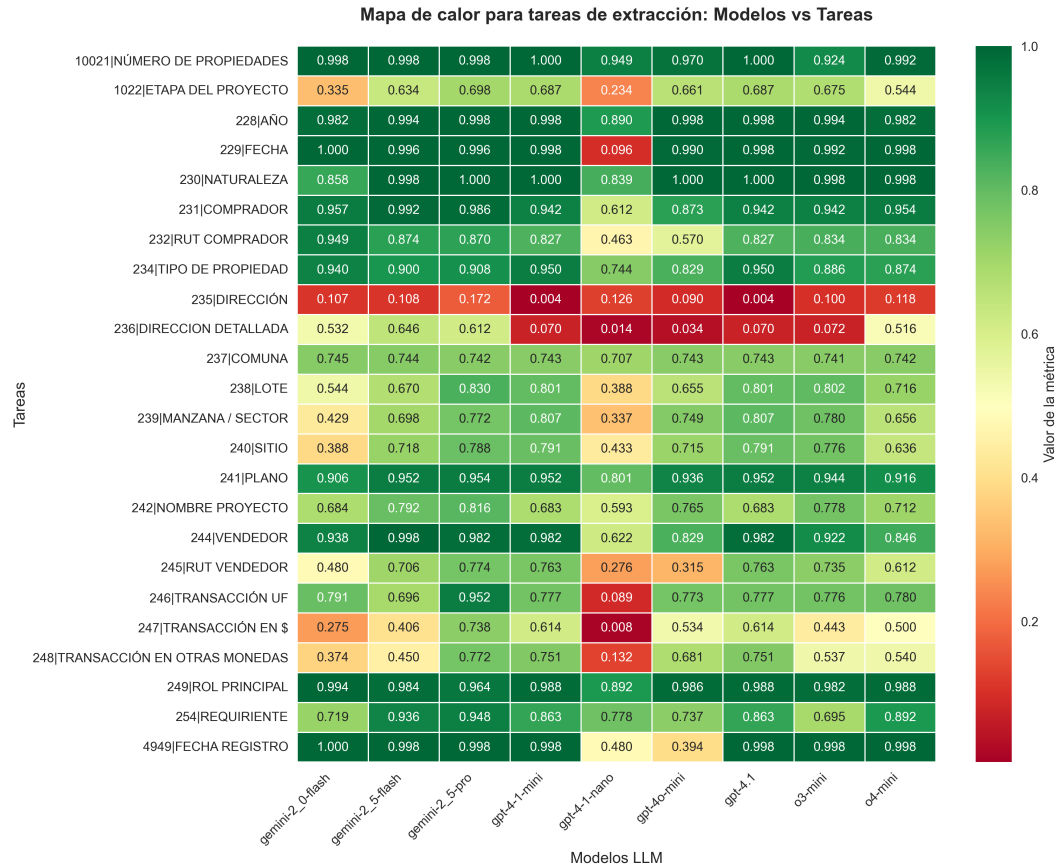


Figura 11: Mapa de calor de exactitud en tareas de extracción, Fuente: Elaboración propia.

Del mapa de calor de exactitud (Figura 11) se pueden extraer las siguientes observaciones clave:

Rendimiento General Elevado: La mayoría de los modelos presentan exactitud superior a 0.9 en tareas estructuradas como extracción de fechas y naturaleza de transacciones. Sin embargo, este alto rendimiento puede deberse a la naturaleza templática de los documentos del CBR, lo que limita la generalización a documentos legales más diversos.

Identificación de Modelos de Bajo Rendimiento: El *benchmark* logra diferenciar claramente los modelos con menor capacidad. El modelo *gpt-4.1-nano* muestra un rendimiento significativamente inferior en múltiples tareas que requieren un formato estructurado y normalización, como la extracción de RUTs (“232| RUT COMPRADOR”, “245| RUT VENDEDOR”) y montos (“247| TRANSACCIÓN EN \$”).

Identificación de Limitaciones del *Benchmark*: El bajo rendimiento en tareas de extracción de direcciones (valores <0.126) revela una limitación metodológica importante. La comparación por similitud coseno no captura equivalencias semánticas evidentes para expertos legales (ej: ".Av. Libertador" vs ".Avenida Libertador"). Esto sugiere la necesidad de métricas de evaluación más sofisticadas o el uso de LLMs como evaluadoras de la comparación (LLM-as-judge).

4.2. Análisis de Resultados en Tareas de Revisión

4.2.1. Rendimiento Balanceado: Métrica F1 (Presente)

El puntaje *F1* para la clase “Presente” ofrece una visión equilibrada entre *precisión* y *recall*, indicando qué tan bien un modelo identifica un requisito como existente sin generar falsos positivos.

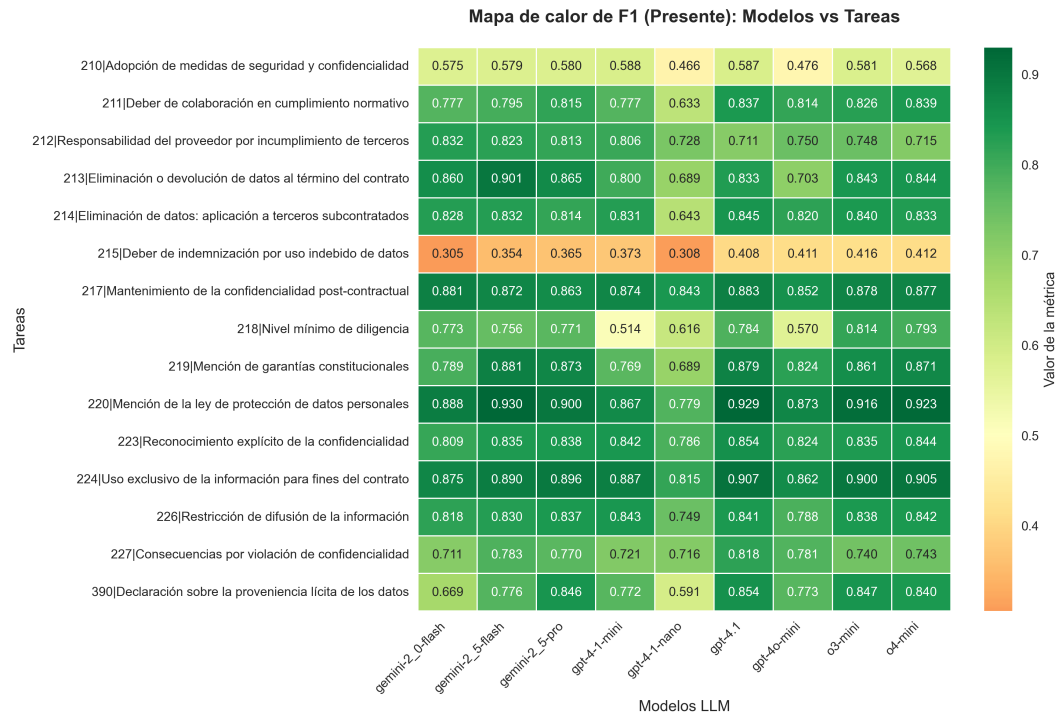


Figura 12: Mapa de calor de F1 (Presente) en tareas de revisión, Fuente: Elaboración propia.

Los modelos *gpt-4.1* y *o4-mini* destacan en cuanto a rendimiento general, manteniendo puntajes F1 consistentemente altos (superiores a 0.8) en la mayoría de las tareas.

La tarea “215| *Deber de indemnización por uso indebido de datos*” resulta ser la más desafiante para todos los modelos, con puntajes F1 bajos en todos los modelos.

Esto se puede deber a tres motivos principalmente:

- La redacción del requisito requiere de modificaciones que añadan más contexto o ejemplos en la medida de lo posible.
- La manera en la que se extraen los fragmentos relacionados al requisito desde la base vectorial no funciona de manera correcta para este requisito (por lo que se tendría que replantear la manera en la que se extraen los fragmentos).

- Se requiere de más fragmentos en la búsqueda vectorial (actualmente se extraen 2) ya que posiblemente este requisito requiere de más contexto para ser identificado correctamente.

4.2.2. Mitigación de Riesgo: Métrica de Precisión (Presente)

La precisión para la clase “Presente” es quizás la métrica más crítica desde una perspectiva de negocio, ya que un valor bajo implica una alta tasa de falsos positivos (el modelo afirma que un requisito existe cuando no es así), lo que podría llevar a la aprobación de un contrato no conforme.

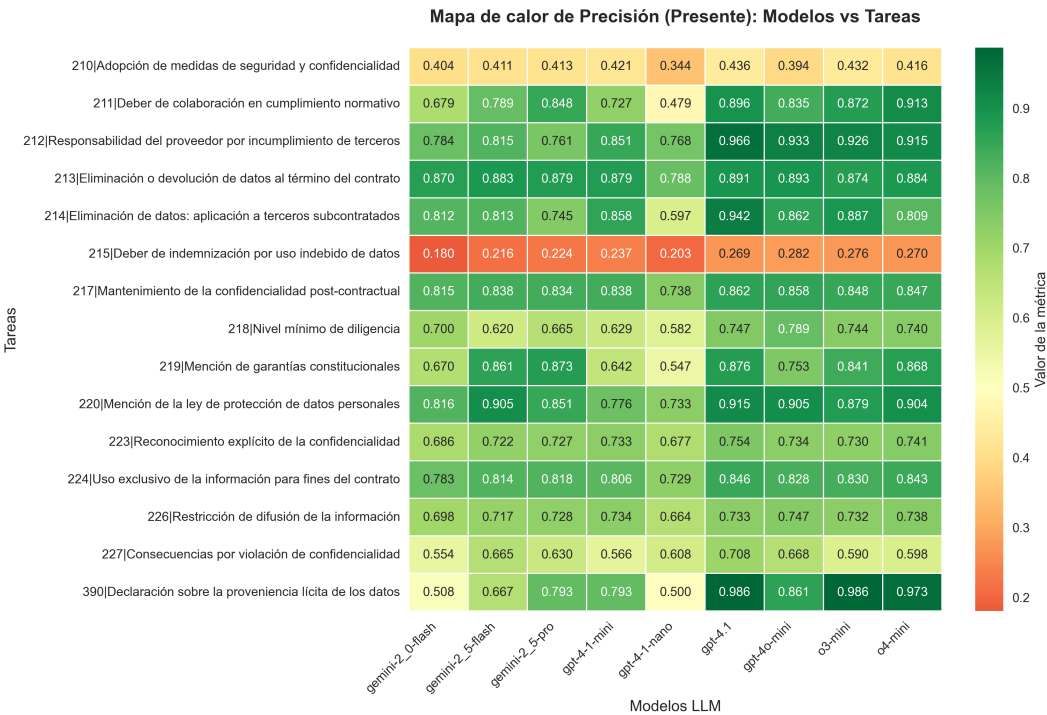


Figura 13: Mapa de calor de Precisión (Presente) en tareas de revisión, Fuente: Elaboración propia.

Los modelos *gpt-4.1*, *o3-mini* y *o4-mini* demuestran los mejores resultados, con valores cercanos a 1.0 en muchas tareas.

En contraste, modelos como *gemin-2.0-flash* y *gpt-4.1-nano* muestran una precisión considerablemente más baja en varias tareas, generando un mayor número de falsos positivos.

4.2.3. Capacidad de Detección: Métricas de Recall (Presente y Ausente)

El recall mide la capacidad del modelo para encontrar todas las instancias relevantes. El *Recall (Presente)* mide si el modelo encuentra un requisito cuando este existe, mientras que el *Recall (Ausente)* mide si detecta correctamente la ausencia de un requisito.

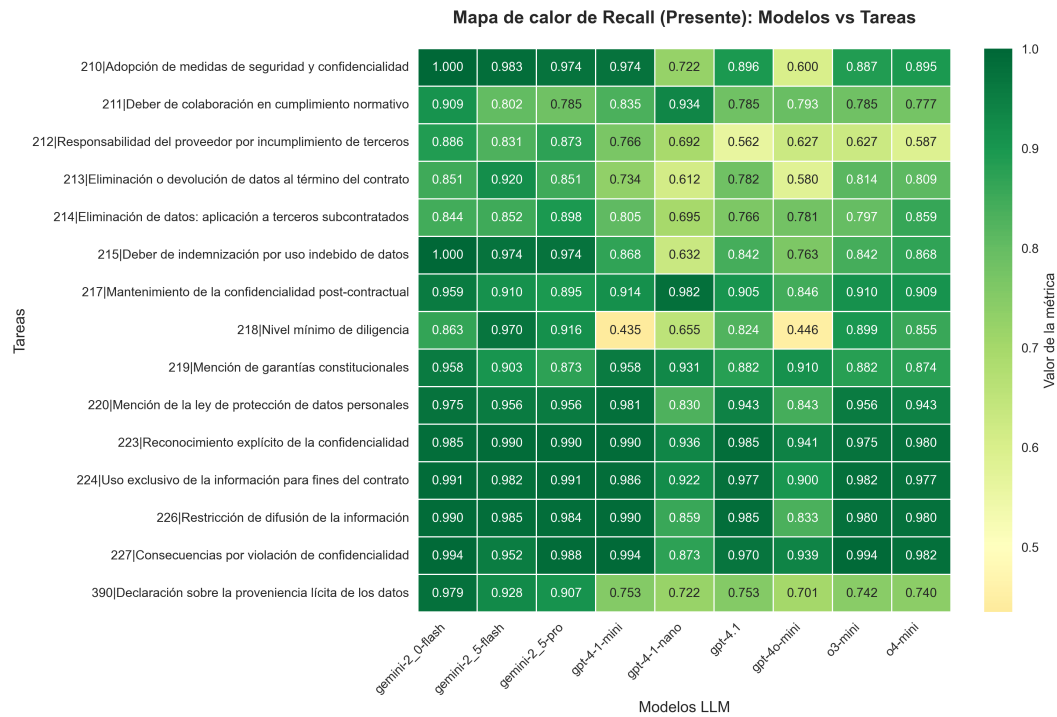


Figura 14: Mapa de calor de Recall (Presente) en tareas de revisión, Fuente: Elaboración propia.

La mayoría de los modelos exhiben un *Recall (Presente)* alto, lo que significa que destacan en encontrar un requisito cuando este está presente en el documento.

Por otro lado, para el caso de *Recall (Ausente)* los resultados son más variados, donde los modelos *gpt-4.1*, *o3-mini* y *o4-mini* sobresalen.

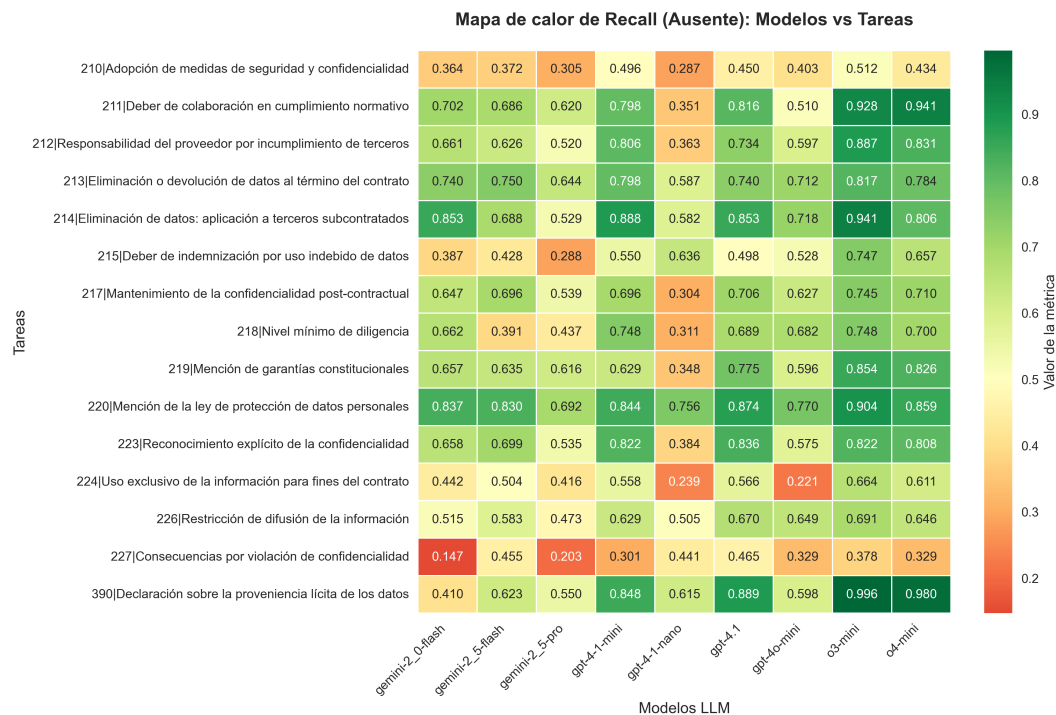


Figura 15: Mapa de calor de Recall (Ausente) en tareas de revisión, Fuente: Elaboración propia.

Se observa que ciertas tareas presentan una mayor dificultad por sobre otras, como “227| Consecuencias por violación de confidencialidad” y “210| Adopción de medidas de seguridad y confidencialidad” muestran un bajo *Recall (Ausente)* a lo largo de todos los modelos, esto se puede deber a que los modelos tienden a confundir parte del contexto que envuelve a estos requisitos con la presencia/incompletitud de estos, produciendo un mayor número de falsos positivos. Para solucionar este problema se pueden hacer uso de un único fragmento del documento además de mejorar el prompt relacionado al requisito con un enfoque en especificidad y detalle de este.

4.3. Síntesis de métricas

Finalmente, se presenta un mapa de calor que resume el rendimiento de todos los modelos evaluados a lo largo de todas las métricas consideradas a evaluar.

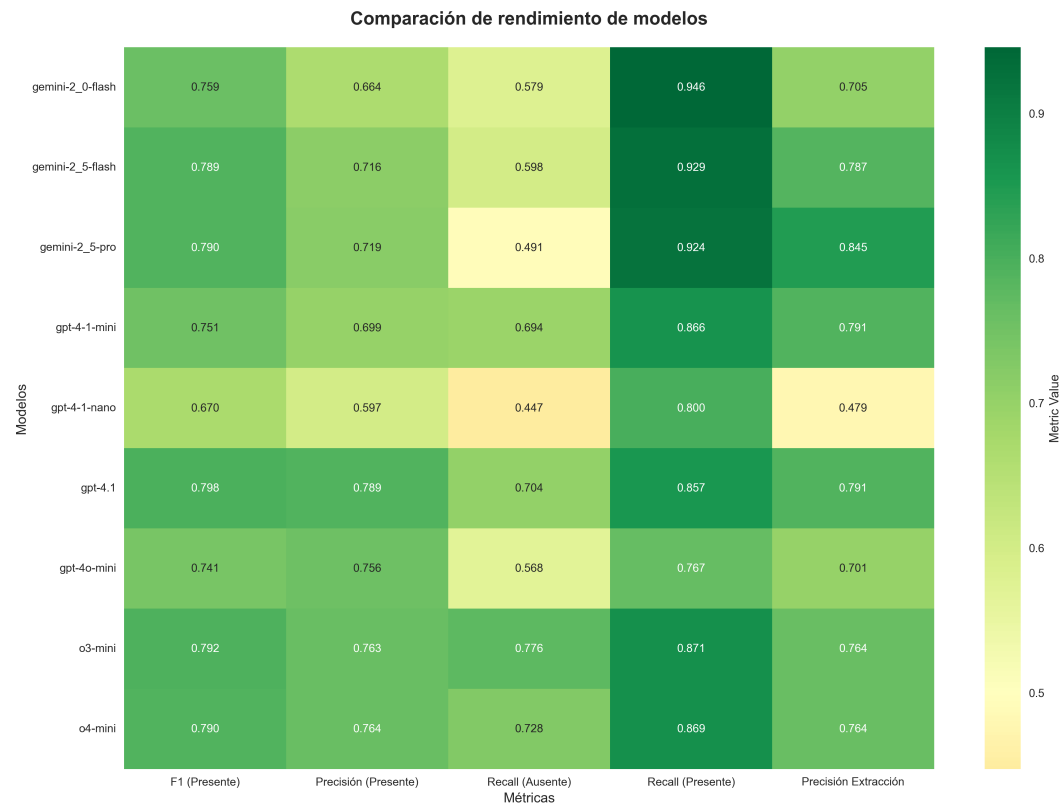


Figura 16: Comparación comprehensiva del rendimiento de todos los modelos, Fuente: Elaboración propia.

El análisis comparativo revela patrones claros de especialización entre modelos:

Para tareas de extracción: *gemini-2.5-pro*

Para tareas de revisión: *gpt-4.1* y *o3-mini*

CAPÍTULO 5

CONCLUSIONES

Esta investigación propuso el diseño e implementación de un *benchmark* sistemático para evaluar y comparar el rendimiento de LLMs en tareas de revisión y extracción de información sobre documentación legal chilena, en el contexto específico de las operaciones de la empresa *Asesorame SpA* y su plataforma *FAVRAS*.

5.1. Validación del Marco Propuesto

Los resultados obtenidos confirman que el *benchmark* desarrollado cumple satisfactoriamente su objetivo principal. La herramienta demostró eficacia para diferenciar las capacidades de diversos modelos de las familias *Gemini* y *GPT*, proporcionando resultados cuantificables mediante métricas relevantes como *Exactitud*, *Precisión*, *Recall* y *F1-Score*. Además, se logró identificar la especialización de ciertos modelos para distintos tipos de tareas.

5.2. Resultados y Limitaciones

Especialización por tipo de tarea: El análisis reveló diferenciación clara entre modelos según el tipo de tarea.

Identificación de limitaciones metodológicas: El bajo rendimiento transversal en tareas específicas expuso limitaciones fundamentales en las estrategias de evaluación. La comparación por similitud coseno no captura equivalencias semánticamente obvias para expertos legales, evidenciando la necesidad de métricas más sofisticadas o enfoques de evaluación basados en *LLM-as-judge*.

Sensibilidad a la ingeniería de prompts: Los resultados demostraron que ciertos requisitos involucrados en las tareas más complejas, requieren refinamiento específico en su formulación o en la estrategia de *RAG*, particularmente aquellos relacionados con obligaciones de indemnización y consecuencias por violación de confidencialidad.

Alcance de dominio restringido: El corpus se limitó a contratos de proveedores y escrituras de compraventa del *CBR*, lo que restringe la generalización a todo el espectro del derecho contractual chileno.

5.3. Trabajo Futuro

Las limitaciones identificadas definen múltiples vías para investigación y desarrollo futuro:

Refinamiento metodológico urgente:

- Implementación de métricas de evaluación semántica más sofisticadas, posiblemente utilizando *LLM-as-judge* para tareas de extracción

Expansión sistemática del corpus:

- Incorporación de contratos laborales, arrendamientos, escrituras societarias, mandatos judiciales, entre otros.
- Diversificación geográfica incluyendo documentos de diferentes conservadores y juzgados.

Automatización de ingeniería de prompts:

- Exploración de *frameworks* como *DSPy*⁶ para optimización automática de instrucciones y ejemplos
- Desarrollo de estrategias adaptativas de fragmentación según requisito

Evaluación continua del ecosistema:

- Análisis comparativo de modelos *open-source* especializados en dominios legales
- Investigación de modelos *fine-tuned* específicamente con corpus legales chilenos

⁶<https://dspy.ai/>

REFERENCIAS BIBLIOGRÁFICAS

- [Chalkidis *et al.*, 2021] Chalkidis, I., Fergadiotis, M., y Androutsopoulos, I. (2021). Multieurlex - A multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. *CoRR*, abs/2109.00904.
- [Chalkidis *et al.*, 2022] Chalkidis, I., Jana, A., Hartung, D., Bommarito, M., Androutsopoulos, I., Katz, D. M., y Aletras, N. (2022). Lexglue: A benchmark dataset for legal language understanding in english.
- [Fei *et al.*, 2023] Fei, Z., Shen, X., Zhu, D., Zhou, F., Han, Z., Zhang, S., Chen, K., Shen, Z., y Ge, J. (2023). Lawbench: Benchmarking legal knowledge of large language models.
- [Guha *et al.*, 2023] Guha, N., Nyarko, J., Ho, D. E., Ré, C., Chilton, A., Narayana, A., Chohlas-Wood, A., Peters, A., Waldon, B., Rockmore, D. N., Zambrano, D., Talisman, D., Hoque, E., Surani, F., Fagan, F., Sarfaty, G., Dickinson, G. M., Porat, H., Hegland, J., Wu, J., Nudell, J., Niklaus, J., Nay, J., Choi, J. H., Tobia, K., Hagan, M., Ma, M., Livermore, M., Rasumov-Rahe, N., Holzenberger, N., Kolt, N., Henderson, P., Rehaag, S., Goel, S., Gao, S., Williams, S., Gandhi, S., Zur, T., Iyer, V., y Li, Z. (2023). Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models.
- [Hendrycks *et al.*, 2021] Hendrycks, D., Burns, C., Chen, A., y Ball, S. (2021). Cuad: An expert-annotated nlp dataset for legal contract review.
- [Lippi *et al.*, 2018] Lippi, M., Palka, P., Contissa, G., Lagioia, F., Micklitz, H., Sartor, G., y Torroni, P. (2018). CLAUDETTE: an automated detector of potentially unfair clauses in online terms of service. *CoRR*, abs/1805.01217.
- [Liu *et al.*, 2023] Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., y Liang, P. (2023). Lost in the middle: How language models use long contexts.
- [Miles y Sunstein, 2007] Miles, T. J. y Sunstein, C. R. (2007). The new legal realism.
- [Nallapati y Manning, 2008] Nallapati, R. y Manning, C. D. (2008). Legal docket classification: Where machine learning stumbles. En Lapata, M. y Ng, H. T., editores, *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 438–446, Honolulu, Hawaii. Association for Computational Linguistics.
- [Naveed *et al.*, 2023] Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., y Mian, A. (2023). A comprehensive overview of large language models.
- [Niklaus *et al.*, 2023] Niklaus, J., Matoshi, V., Rani, P., Galassi, A., Stürmer, M., y Chalkidis, I. (2023). Lextreme: A multi-lingual and multi-task benchmark for the legal domain. En *Findings of the Association for Computational Linguistics: EMNLP 2023*, p. 3016–3054. Association for Computational Linguistics.

- [Nikolaos Aletras, et al., 2016] Nikolaos Aletras, et al. (2016). Predicting judicial decisions of the European Court of Human Rights: a Natural Language Processing perspective.
- [Santosh et al., 2024] Santosh, T. Y. S. S., Weiss, C., y Grabmair, M. (2024). Lexsumm and lext5: Benchmarking and modeling legal summarization tasks in english.
- [Tuggener et al., 2020] Tuggener, D., von Däniken, P., Peetz, T., y Cieliebak, M. (2020). LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. En Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., y Piperidis, S., editores, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 1235–1241, Marseille, France. European Language Resources Association.
- [Wang et al., 2023] Wang, S. H., Scardigli, A., Tang, L., Chen, W., Levkin, D., Chen, A., Ball, S., Woodside, T., Zhang, O., y Hendrycks, D. (2023). Maud: An expert-annotated legal nlp dataset for merger agreement understanding.
- [Wu et al., 2023] Wu, S., Irsoy, O., Lu, S., Dabrowski, V., Dredze, M., Gehrmann, S., Kambadur, P., Rosenberg, D., y Mann, G. (2023). Bloomberggpt: A large language model for finance.
- [Zheng et al., 2021] Zheng, L., Guha, N., Anderson, B. R., Henderson, P., y Ho, D. E. (2021). When does pretraining help? assessing self-supervised learning for law and the casehold dataset. *CoRR*, abs/2104.08671.