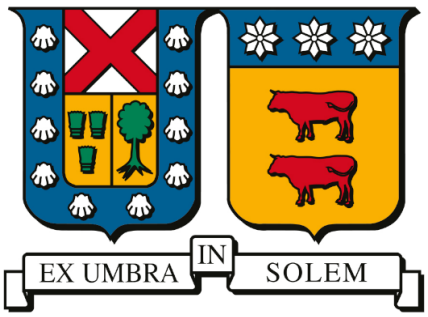
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO INGENIERIA COMERCIAL
SANTIAGO - CHILE



TÍTULO DE MEMORIA (EL TÍTULO SÓLO PUEDE TENER UN MÁXIMO DE 3 LÍNEAS)

SANTIAGO JESÚS VASCONCELLO ACUÑA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO COMERCIAL

PROFESOR GUÍA	:	SR. PABLO ISLA
PROFESOR CORREFERENTE	:	SR. THIERRY DE SAINT PIERRE.

Diciembre 2023

A mi familia . . .

Puede ocupar este espacio para escribir una dedicatoria (opcional). [Revise el archivo maestro `memoria.tex` para modificar / eliminar esta sección.]

(AGRADECIMIENTOS) [Título es opcional]

Agradezco a quienes contribuyeron para ir mejorando esta plantilla hecha en LaTeX.

Los aportes y comentarios de distintas personas en el Departamento de Industrias fueron muy útiles para que este documento puede ser ocupado para mejorar la presentación de tesis y memorias del Departamento (y la universidad).

Para el impaciente ...

Por favor ocupar `git` :

```
git clone https://github.com/jaimercz/utfsm-thesis.git
```

Para los interesados en `git` revisar [**Benoit2018**].

Abra el archivo de configuración `config.tex` para cambiar título, autor, fecha, etc. de la portada y del documento en general.

Abra y compile el documento maestro `memoria.tex` .

```
$ pdflatex memoria.tex
$ biber memoria
$ pdflatex memoria.tex
$ pdflatex memoria.tex
```

Esta version ocupa `biber` en lugar de `natbib / bibtex` :

Si hay errores, verifique primero que todos los paquetes LaTeX han sido instalados.

Si desea omitir alguna sección (dedicatoria, agradecimientos, etc.), revise el documento maestro `memoria.tex` y agregue o comente (o elimine) las líneas correspondientes.

Por ejemplo, para eliminar esta sección, borre las líneas:

`memoria.tex` (extracto)

```
\section*{Agradecimientos}
\insertFile[plain]{agradecimientos}}
```

RESUMEN EJECUTIVO

Plantilla LaTeX para las Memorias y Tesis del Departamento de Ingenieria Comercial, UTFSM.

Se incluyen también algunos ejemplos de cómo incorporar tablas y gráficos en distintas presentaciones respetando las Normas de Biblioteca para Memorias y Tesis de la UTFSM.

Palabras Clave. LaTeX, Plantilla para Memoria, Departamento de Ingenieria Comercial, UTFSM.

¡Importante! [LEAME]

Impresión por un solo lado.

A partir del año 2016, el Departamento de Ingenieria Comercial sólo requiere la entrega digital de los archivos de memorias y tesis. Por este motivo, este documento está preparado para ser impreso por un solo lado de una hoja (*"oneside"*), y facilitar así su lectura en pantallas. Esta configuración es parte de archivo de clase `thesis_utfsm.cls` .

Codificación de caracteres.

Todos los archivos `*.tex` de esta plantilla han sido preparados ocupando la codificación de caracteres por defecto *unicode* (UTF-8). Windows (y algunas versiones de OSX) ocupan otro tipo de codificación (ej. *Windows-1252* o *Mac Roman*).

Si deseas ocupar esta plantilla y encuentras problemas con los caracteres acentuados, entonces puedes optar por una de estas tres alternativas:

i) cambiar tu editor (TexMaker, TexStudio, TexShop, etc.) para que ocupe UTF-8 como codificación de caracteres por defecto; o

ii) cambiar la codificación de cada documento `*.tex` para que ocupe la codificación nativa de tu sistema operativo; y, modificar el archivo `config.tex` la línea que dice:

   OSX, Linux: `\usepackage[utf8x]{inputenc}`

   Windows: `\usepackage[latin1]{inputenc}`

   Overleaf: `\usepackage[utf8]{inputenc}` https://overleaf.com

iii) escribir todo ocupando caracteres pre-acentuados (ej. `\'a` en lugar de á).

Recuerde:

Mezclar documentos de distintas codificaciones puede generar muchos problemas al momento de compilar.

ABSTRACT

This is a LaTeX thesis template for the Departamento de Ingenieria Comercial, UTFSM. A few examples about the inclusion of figures and tables are also provided.

(The abstract can be edited by opening the file `includes/abstract.tex` .)

Keywords. LaTeX, Thesis Template, Departamento de Ingenieria Comercial, UTFSM

Instrucciones para la Plantilla.

Editar el archivo `/includes/abstract.tex` para modificar los contenidos de esta sección.

Si no desea incluir un abstract, editar el archivo `/memoria.tex` , y comentar o borrar la sección que se muestra a continuación.

`/memoria.tex` (extracto)

```
\section*{ABSTRACT}
\insertFile[plain]{abstract}                    % Archivo abstract.tex
```

Índice de Contenidos

Índice de Tablas

Índice de Figuras

1 | Introducción

La inteligencia artificial, también conocida como IA, ha experimentado un notable auge en la industria en los últimos tiempos de la mano de la llamada Industria 4.0 [5], especialmente en el ámbito en areas algo reacias como la administración y finanzas[4]. Este incremento no se debe necesariamente a un aumento en la capacidad de cómputo, ya que esta ha ido creciendo gradualmente a lo largo del tiempo (buscar respaldo). Anteriormente, aunque importante, no generaba tanto interés como en la actualidad. No fue sino hasta que OpenAI lanzó ChatGPT el 30 de noviemnbre de 2022 que el público en general pudo experimentar, probar y comprender de manera más completa la gran revolución llamada inteligencia artificial generativa [12].

De acuerdo con Google, "La inteligencia artificial generativa se refiere al uso de la IA para crear contenido, como texto, imágenes, música, audio y videos"[1]. Gracias a su interfaz amigable, resultó sencillo para personas de diversas industrias descubrir que existía una herramienta capaz de generar texto y responder preguntas de manera comprensible, incluso para aquellos que no eran expertos en tecnología.

La génesis de esta tesis se basa en la experiencia de llevar a cabo un proyecto utilizando estas tecnologías y los riesgos asociados a ellas. En este contexto, entendemos el riesgo como cualquier aspecto que pueda afectar tanto al equipo involucrado en la creación del proyecto como a los resultados obtenidos. El proyecto se centró en el uso de un modelo de lenguaje de gran envergadura, conocido como LLM por sus siglas en inglés, limitándose a la generación de texto. Por lo tanto, no profundizaremos en otros tipos de inteligencia artificial generativa, como la generación de imágenes o audio. El enfoque principal de este trabajo se concentra solo en el area del procesamiento del lenguaje natural aplicados a LLM.

1.1. Obejetivos

1.1.1. Objetivo General

Determinar los factores de riesgo que pueden llegar a influir, tanto de la creación como del uso de aplicaciones que utilicen modelos grandes de lenguaje (LLM) aplicados a la industria, usando de base el proyecto de búsqueda de jurisprudencia de los tribunales ambientales.

1.1.2. Objetivo Específico

1. Determinar una posible estructura de una aplicación usando LLM
2. Desarrollar los estados del Arte del uso de LLM y de los modelos generativo en si
3. Desarrollar los problemas que conlleva el uso de información para alimentar dichos problemas
4. Analizar un proceso de ETL de principio a fin para observar sus posibles riesgos

1.2. Metodologia

La metodología empleada en esta tesis se estructura en torno a tres componentes esenciales: la creación del proyecto, un ejemplo de uso concreto y la evaluación de los riesgos asociados a cada etapa del proceso, tanto en la fase de desarrollo del proyecto como en su aplicación práctica.

1.2.1. Creación del Proyecto

Esta fase inicial comprende el desarrollo del proyecto basado en IA generativa. Incluye los siguientes pasos:

- **Definición de Objetivos y Alcance:** Establecimiento claro de los propósitos y límites del proyecto, identificando las metas a alcanzar.
- **Selección de Tecnologías y Herramientas:** Evaluación y elección de las tecnologías y herramientas apropiadas para la implementación del proyecto.
- **Diseño de la Arquitectura:** Desarrollo de la estructura y componentes del proyecto, considerando aspectos de escalabilidad y rendimiento.
- **Implementación y Desarrollo:** Construcción efectiva del proyecto, incluyendo la programación y configuración de la inteligencia artificial generativa.

1.2.2. Ejemplo de Uso del Proyecto

Esta fase implica la aplicación práctica del proyecto en un contexto específico, demostrando su funcionalidad y utilidad. En este caso nuestro interés mas que en el output que genere la aplicación, es como funciona internamente el proceso cosa que para la siguiente etapa sea más fácil

1.2.3. Evaluación de Riesgos

Se trabajará la identificación y análisis de los riesgos potenciales en cada etapa del proceso, así como los riesgos derivados del caso de uso. Incluye:

- **Riesgos en la Creación del Proyecto:** Identificación de posibles obstáculos y contratiempos durante la etapa de concepción y desarrollo.
- **Riesgos en el Uso de la Aplicación:** Consideración de los riesgos asociados a la implementación práctica del proyecto en el contexto definido.

Esta metodología proporciona un enfoque integral para la creación y aplicación de un proyecto utilizando LLM, permitiendo una evaluación de los riesgos en cada etapa del proceso y en el caso de uso específico. Esto facilita la toma de decisiones informadas y la formulación de estrategias para mitigar posibles contratiempos.

2 | Creación del Proyecto

Los Tribunales Ambientales son órganos jurisdiccionales especiales, sujetos a la superintendencia directiva, correccional y económica de la Corte Suprema, cuya función es resolver las controversias medioambientales de su competencia y ocuparse de los demás asuntos que la ley somete a su conocimiento [10]. Estos tribunales generan una cantidad de jurisprudencia que puede ser encontrada en su portal de consulta llamado buscador ambiental [3].

El proyecto consiste en la generación de un chatbot en donde se pueda preguntar sobre la jurisprudencia de estos tribunales, aunque por razones de capacidad el chatbot se vea acotado solamente a las reclamaciones recibidas por el tribunal.

Por lo que, este proyecto consiste en un proceso de extracción de datos desde el buscador ambiental, transformación de estos datos para su utilización, generación de vectores de estos datos para que puedan interaccionar con la aplicación, carga de estos en una base de datos, para que después la aplicación pueda interactuar con ellos y mandando esa información a el LLM, siendo en este caso gpt-4 perteneciente a OpenAI.

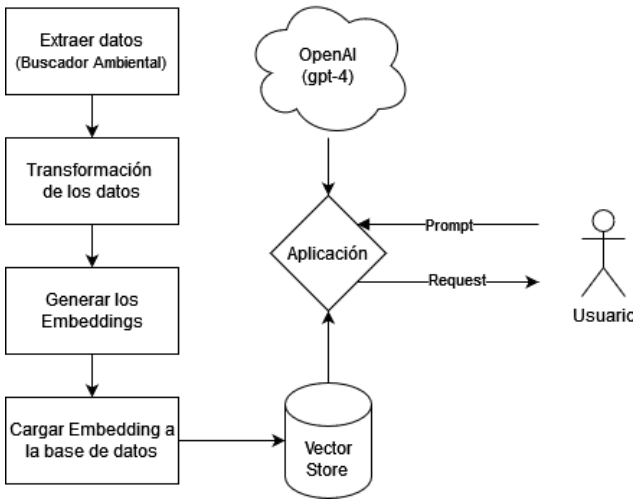


Figura 2.1: Estructura basica de la aplicación
(Fuente: Elaboración propia)

A partir de esta estructura mientras se avance en el desarrollo, se explicará parte por parte el proceso y con ello los riesgos de cada uno de ellos.

2.1. ETL

Para realizar el proyecto fue necesario realizar un proceso de ETL. El término ETL se refiere a las técnicas de .Extracción, Transformación y Carga"(Extract, Transform, Load), que constituyen un proceso clave para los datos necesarios para el proyecto. Este proceso implica la extracción de datos de fuentes heterogéneas, su transformación para

ajustarse a las necesidades del negocio y su posterior carga en un destino que, por lo general, es un almacén de datos diseñado para el análisis y la generación de informes[2].

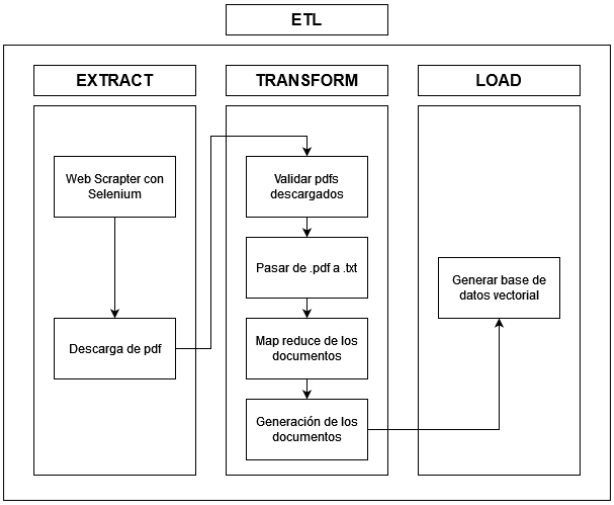


Figura 2.2: Estructura del proceso de ETL para el Buscador Ambiental
(Fuente: Elaboración propia)

La fase de extracción implica la recolección de datos de múltiples fuentes, que pueden variar desde bases de datos estructuradas hasta información no estructurada en la web. La transformación se refiere al proceso de limpieza, conversión, y consolidación de estos datos en un formato adecuado para el análisis. Finalmente, la carga es el proceso de transferir los datos transformados al sistema de destino, donde se pueden almacenar y utilizar para la toma de decisiones estratégicas [2].

2.1.1. Extract

La información requerida para el desarrollo del Chatbot se obtuvo del "Buscador ambiental"del Tribunal de Protección Ambiental de Chile a través de su sitio web[3]. Este portal aloja todos los documentos públicos disponibles para su consulta en cualquiera de los tres tribunales ambientales. Para acceder a la base de datos necesaria, se llevó a cabo la creación de un bot capaz de recopilar las entradas de este buscador de manera análoga a un usuario convencional.

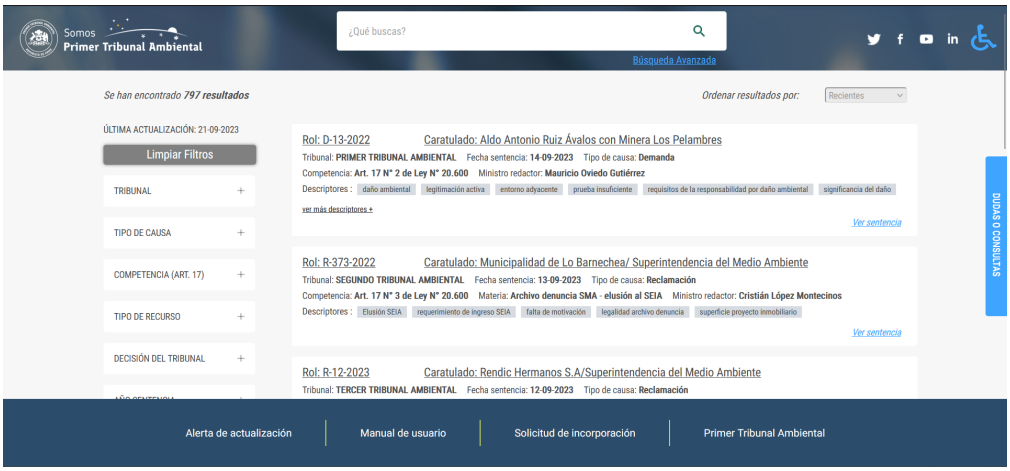


Figura 2.3: Buscador de la pagina del Primer Tribunal Ambiental
(Fuente: Pagina del Primer Tribunal Ambiental)

Para esta tarea, se empleó Selenium, una herramienta originalmente diseñada para generar pruebas, pero que, debido a la naturaleza reactiva y dinámica de los sitios web, así como a la detección de bots por parte de algunas páginas, resultó ser la elección más apropiada. Este bot, después de explorar todas las páginas del buscador ambiental, como se ilustra en la Figura 2.3, logró recuperar cada uno de los enlaces individuales que conducen a las páginas específicas de cada caso, tal como se muestra en la Figura 2.4.

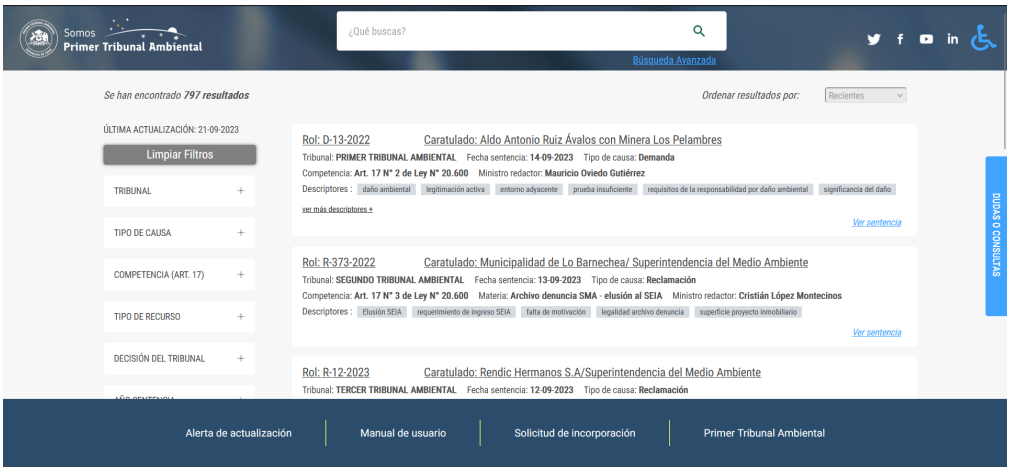


Figura 2.4: Screenshot de una sola reclamación en la pagina web del buscador ambiental
(Fuente: Pagina del Primer Tribunal Ambiental)

Posteriormente, se contemplaba la posibilidad de obtener tanto los enlaces a cada documento en formato PDF como la información detallada de cada uno de estos documentos mediante la creación de un nuevo bot. Sin embargo, durante el proceso de desarrollo de este bot, se logró acceder a la API que permitía obtener directamente todos los datos mencionados anteriormente. Esto suprimió la necesidad de crear otro tipo de bot utilizando Selenium, ya que bastaba con realizar una solicitud a la mencionada API.

Para completar la fase de extracción de datos (Extract), una vez que se había obtenido toda la información mediante las solicitudes a la API, el último paso consistió en generar nuevas solicitudes con el objetivo de descargar todos los archivos PDF de cada una de las entradas. Estos archivos ahora están descargados y listos para la próxima etapa del proyecto, que implica la transformación de los datos con el fin de obtener la información necesaria para construir la base de datos a partir de los documentos.

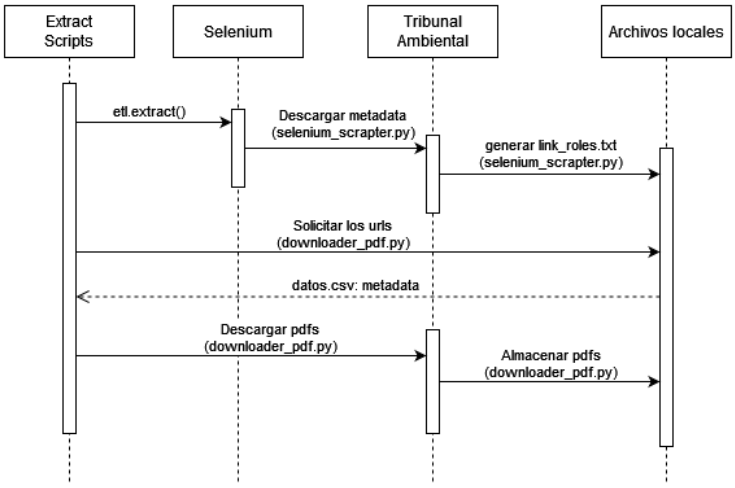


Figura 2.5:
(Fuente: Elavoración propia)

2.1.2. Transform

En la continuación en el proceso de ETL (Extracción, Transformación y Carga), los PDFs que previamente han sido descargados requieren ser sometidos a modificaciones con el objetivo de convertir la información que inicialmente se presenta en un estado "sucio" en datos "limpios" que puedan ser adecuadamente utilizados en el proyecto. Este proceso se denomina "transformación", o "transform," en inglés.

Entre los datos descargados, nos encontramos con un extenso número de PDFs que presentan dificultades significativas para su manipulación. Esto se debe a que el Tribunal Ambiental no sigue un formato estándar en la estructura de las reclamaciones presentadas. En consecuencia, cada uno de los textos posee un formato propio, lo que complica en gran medida la extracción eficiente de las diversas secciones contenidas en dichos textos. Sin embargo, gracias al funcionamiento del proceso de semejanza semántica, esta diversidad de formatos no representa un problema insuperable para el proyecto.

No obstante, surgen dificultades adicionales cuando se trata de las reclamaciones que son presentadas a los tribunales ambientales en formato digital o, en su defecto, en forma de fotocopias. Esto implica que no todos los documentos están habilitados para su procesamiento. En consecuencia, el primer paso en el proceso de transformación involucra la discriminación de qué PDFs son susceptibles de ser procesados y cuáles no. Para llevar a cabo esta tarea, se ha desarrollado un script capaz de detectar texto dentro de un archivo PDF. Si el texto es legible, se almacena; de lo contrario, se elimina.

Una vez separados los PDFs legibles y adecuados para el trabajo posterior, se procede con la transformación de estos documentos al formato TXT (texto plano). Esta etapa se lleva a cabo considerando la conveniencia de trabajar con archivos en formato de texto en comparación con los archivos en formato PDF puro, dado que el próximo método de transformación, que implica el uso de map-reduce en Langchain, requiere que los datos estén en formato de texto.

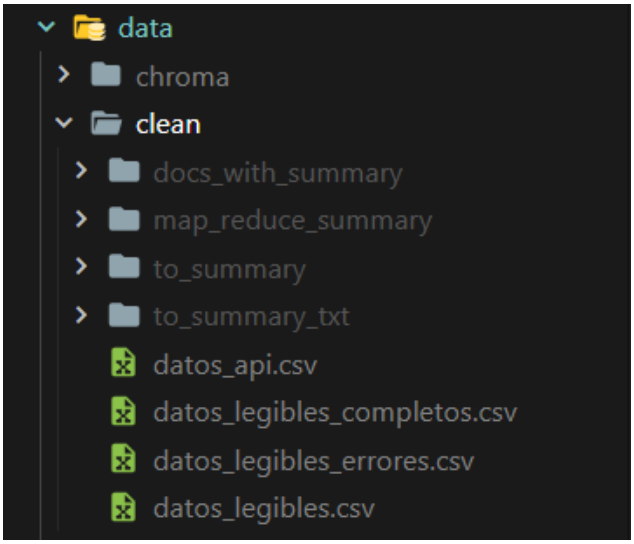


Figura 2.6:
(Fuente: Elavoración propia)

2.1.2.1. Map-Reduce

El proceso de Map-Reduce es un modelo de programación diseñado para procesar grandes cantidades de datos de manera eficiente, escalable y distribuida a través de clústeres de servidores. En el contexto de un archivo PDF muy grande, por ejemplo, si se quisiera resumir el contenido o analizar la frecuencia de ciertas palabras, Map-Reduce podría ser utilizado para dividir la tarea en partes más pequeñas y manejables. Primero, la función de map tomaría el texto del PDF y lo dividiría en elementos más pequeños, como párrafos o líneas, asignando a cada uno un resumen

intermedio [14]. Luego, la función de reduce recogería todos los resumenes intermedios asociados con el documento y los combinaría para producir un resultado agregado, con un resumen de todo el documento.

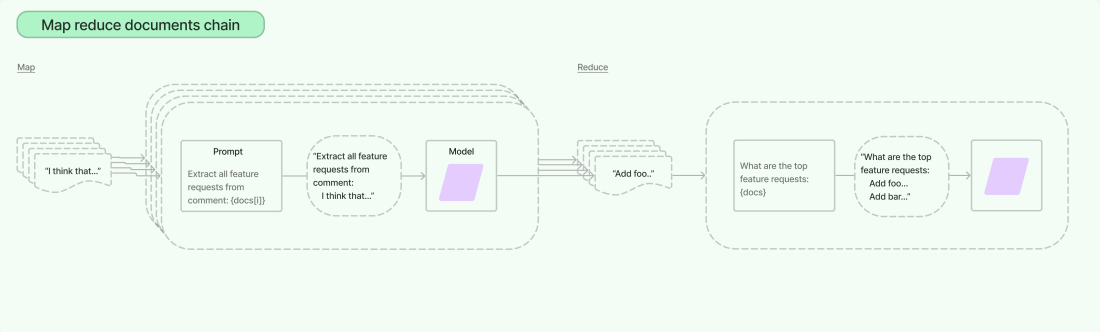


Figura 2.7:
(Fuente: Elavoración propia)

Sin embargo, es importante destacar que un archivo .txt puede contener un número de tokens demasiado elevado como para ser reducido de manera inmediata. En situaciones de este tipo, es necesario recurrir a un proceso de subdivisión que fragmente los textos en segmentos con un número de tokens inferior al límite impuesto por la API de OpenAI. Cada archivo .txt puede ser dividido, resumido y exportado a un nuevo archivo .txt una vez que ha sido fragmentado previamente en segmentos.

Los documentos procesados son combinados utilizando otro proceso de Langchain para obtener un resultado final consolidado. Para concluir el proceso de transformación, los resúmenes generados después de haber pasado por el procedimiento de map-reduce se someten a un último paso antes de ser incorporados en la base de datos. Este paso implica la fusión de los resúmenes con la información obtenida a través de la información Extraida por Selenium previamente, presentada en formato de texto. Este proceso resulta en la creación de un único documento que engloba toda la información, al cual nos referiremos como “documentos finales”. Con esto, se concluye la fase de transformación y se procede al último procedimiento, conocido como “carga” (Load), que consiste en el almacenamiento estos documentos finales en la base de datos.

2.1.3. Load

Al culminar el proceso de Extracción, Transformación y Carga (ETL), resulta fundamental llevar a cabo la fase de carga, también conocida como “load” en inglés, en la cual se incorporan todos los documentos previamente descargados y transformados en una base de datos. Para este proyecto, en el cual se utiliza LangChain, resulta de vital importancia fragmentar los documentos en secciones más pequeñas, por lo que se deben dividir en chunks todos los documentos.

Esta necesidad surge debido a que los documentos deben ser sometidos a un proceso de Embeddings antes de ser introducidos en la base de datos. Esto se debe principalmente a que las funciones de Embeddings tienen un límite en la extensión de grupos de caracteres, conocidos como “tokens”, que pueden ser procesados. En el contexto del modelo de Embeddings “text-embedding-ada-002”, este límite se establece en 8191 tokens [15], lo que constituye la longitud máxima de los fragmentos.

2.1.3.1. Embeddings

Por lo tanto, cuando se trabaja con documentos extensos, es imperativo dividirlos en fragmentos más pequeños antes de proceder con su incorporación. Según la información proporcionada en el Blog de OpenAI, los embeddings son “representaciones numéricas de conceptos convertidos en secuencias numéricas, lo que facilita que las computadoras comprendan las relaciones entre los conceptos”[15]. En términos sencillos, los embeddings son representaciones vectoriales de texto que permiten su comprensión por parte del Modelo de Lenguaje de Gran Tamaño (LLM). Dado que

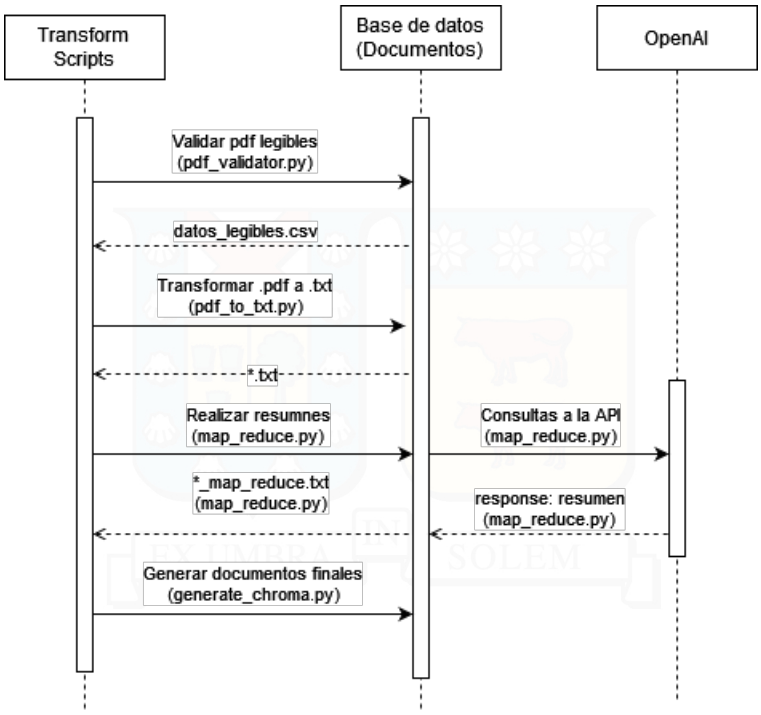


Figura 2.8:
(Fuente: Elavoración propia)

los LLM son redes neuronales, el proceso de Embedding resulta esencial para traducir el texto en números, que es el formato comprensible para esta red neuronal.

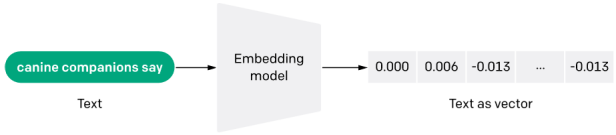


Figura 2.9:
(Fuente: Elavoración propia)

Los embeddings resultantes se almacenan posteriormente en una base de datos vectorial denominada ChromaDB. Esta base de datos ha sido diseñada para ser compacta, escalable y eficiente, con el propósito de almacenar y recuperar vectores de manera efectiva. ChromaDB genera índices que permiten una recuperación rápida y eficiente de los embeddings en función de las consultas realizadas por los usuarios[18].

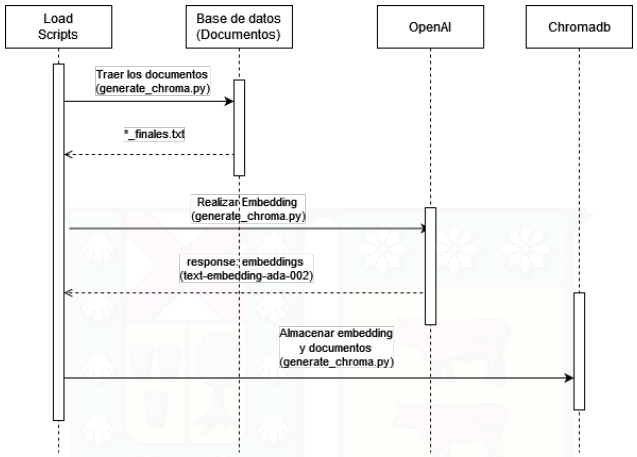


Figura 2.10:
(Fuente: Elavoración propia)

2.2. Chatbot

La estructura del chatbot se desarrolló en su totalidad utilizando el lenguaje de programación Python. Esta elección se debió a la experiencia del equipo en Python, lo que facilitó tanto la creación del frontend como del backend de la aplicación.

Para la parte de la interacción del usuario (frontend), se empleó Python junto con el framework Flask para la presentación de contenido en pantalla, incluyendo tanto la estructura de HTML como las hojas de estilo CSS. Además, se aprovechó la potencia del framework Bootstrap para agilizar el proceso de maquetación.

En cuanto al backend, se implementó una API con el objetivo de facilitar la interacción entre el frontend y el backend. Para este propósito, se utilizó FastAPI, un framework que permite la creación rápida de APIs y que ofrece la ventaja de contar con Swagger para la prueba y generación automática de documentación.

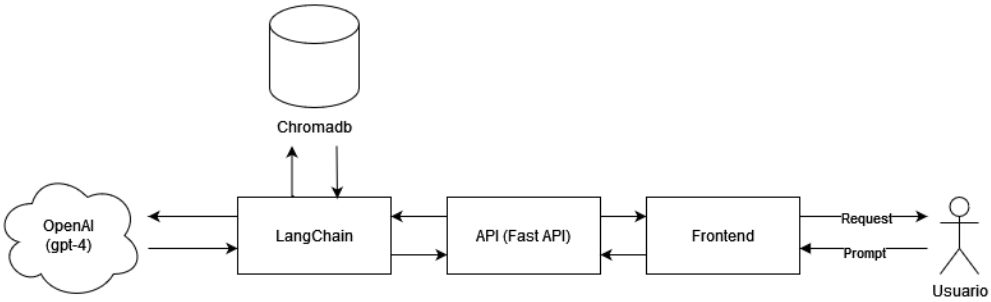


Figura 2.11: Diagrama de funcionamiento del Chatbot
(Fuente: Elavoración propia)

El funcionamiento interno de la aplicación se basa en el framework Langchain para la interacción con el Modelo de Lenguaje Grande (LLM). LangChain es un framework poderoso que simplifica el desarrollo de aplicaciones utilizando modelos de lenguaje grandes (LLM). Proporciona una interfaz única y personalizable capaz de gestionar diferentes LLM, incluida la gestión rápida, el procesamiento, el aumento de datos, la orquestación de agentes, el almacenamiento y la evaluación. Este marco versátil permite a los desarrolladores integrar perfectamente los LLM con sus flujos de trabajo del mundo real y datos con el mínimo esfuerzo [18].

Para la base de datos se utilizó ChromaDB. ChromaDB es una base de datos vectorial sin esquemas diseñada específicamente para su uso en aplicaciones de inteligencia artificial. Es liviano y muy potente, lo que permite el

almacenamiento, la recuperación y la gestión eficiente de datos vectoriales (Embeddings), lo cual es esencial para las aplicaciones de chat de documentos basadas en LangChain y OpenAI [18]. Para que finalmente estas trabajaran en conjunto con los modelos de OpenAI, específicamente el modelo gpt-4 que actualmente es el más potente del mercado.

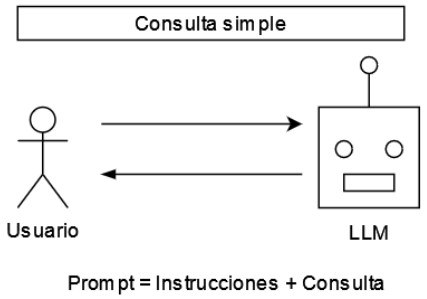


Figura 2.12:
(Fuente: Elavoración propia)

La función principal del chatbot es generar respuestas utilizando la técnica Retriever-Augmented Generation"(RAG), que implica proporcionar contexto adicional en el prompt enviado a OpenAI. Mientras que una generación simple suele constar de instrucciones y una consulta, en RAG se agrega contexto dentro del prompt con el propósito de reducir la probabilidad de alucinaciones y mejorar la calidad de las respuestas.

En cambio, mediante Retriever-Augmented Generationçonsiste al igual que el proceso anterior se entregan las instrucciones y una consulta, pero junto ello se agrega un contexto de este, cosa de que el modelo largo de lenguaje tenga menor capacidad de alucinar y generar una mejor respuesta. Podemos decir que Retriever-Augmented Generation"(RAG) se refiere a un modelo de generación de lenguaje que se mejora con la capacidad de recuperar información de una memoria no paramétrica, como un índice de vectores del buscador ambiental, además de la memoria paramétrica que ya posee. Este enfoque se utiliza para mejorar la generación de respuestas en tareas de procesamiento de lenguaje natural (NLP) que requieren conocimiento intensivo [9].

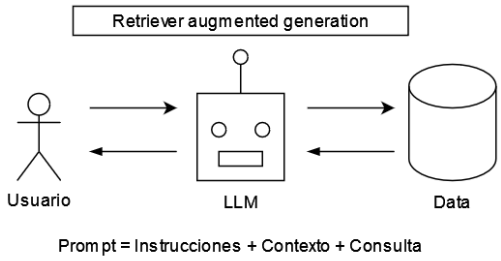


Figura 2.13:
(Fuente: Elavoración propia)

3 | Ejemplo de Uso del Proyecto

4 | Evaluación de Riesgos

4.1. Creación del Proyecto

4.1.1. Sesgos

Los Modelos de Lenguaje Grande (LLM), al ser entrenados con una masiva cantidad de datos, pueden manifestar sesgos debido a la procedencia de los datos utilizados en su entrenamiento. Estos sesgos pueden dar lugar a desafíos cuando se aplican en contextos distintos, ya que las respuestas generadas por el modelo pueden no ser adecuadas ni ajustarse a la realidad de esos nuevos escenarios.

Los modelos preentrenados con corupus generados por humanos contien sesgos sociales hacia ciertos grupos demograficos, estos sesgos son preocupantes, debido a que pueden ser propagados o incluso amplificados en las tareas que estos modelos realizan [7].

Como ejemplo podemos citar el dicho por Bill Gates en su entrevista “Can AI Save the World? Expert Insights with Bill Gates” en donde menciono que: “Los sesgos en los modelos de IA pueden llevar a diagnósticos incorrectos, como se vio en el ejemplo donde Chat GPT diagnosticó erróneamente la tuberculosis como gripe debido a las bajas tasas de tuberculosis en los EE. UU.” [13]. Con lo que podemos dimensional el efecto real de estos sesgos en lo correcto que puede llegar a ser una respuesta por parte de estos modelos.

4.1.2. Elección correcta del modelo

Actualmente la oferta de grandes modelos de lenguaje es muy amplia, desde los privados como: ChatGPT, PaLM, Bloom, etc. Como también modelos de código abierto como: Llama 2, OpenLLaMA, Falcon, Dolly, etc. [**modelo2**] Con sus respectivas variantes, debido a que existen variables del modelo como por ejemplo Llama 2 que se puede encontrar en versión de 7, 13 y 70 billones de parámetros [8].

Elegir un modelo para trabajar es sumamente importante debido a que: la diversidad y calidad de los datos de preentrenamiento influyen sustancialmente en la capacidad del modelo de lenguaje para comprender y proporcionar respuestas precisas, que el tamaño puede tener una gran influencia en el rendimiento, que el soporte lingüistico podría ser crucial dependiendo la necesidad [11].

4.1.3. Volatilidad del Mercado

Hasta la fecha actual, el 06 de noviembre de 2023, la creación de Chatbots utilizando el método RAG se perfilaba como una de las tendencias más destacadas en el mercado, siendo posiblemente una de las aplicaciones más prometedoras de los LLM. No obstante, en este mismo día, durante la OpenAI DevDay Keynote, se anunciaron novedades significativas, como la entrada en escena de los GPTs, que permite personalizar versiones de ChatGPT con instrucciones, conocimiento extra y cualquier otra combinación de habilidades [16]. Además, se introdujeron otros modelos como gpt-4 turbo, un playground de desarrollo para la herramienta, text-to-speech (TTS), entre otros [17].

En este contexto, comprometerse con cualquier tecnología conlleva riesgos, especialmente en este período caracterizado por una volatilidad extrema y una inversión extremadamente agresiva en inteligencia artificial. La Inteligencia Artificial Generativa continúa evolucionando de manera acelerada, lo que la hace cada vez más disruptiva y más eficiente. Por lo tanto, la investigación y la implementación de soluciones de inteligencia artificial centradas en asistentes o chatbots representan un compromiso de alto riesgo en este entorno en constante cambio.

4.2. Uso de la Aplicación

4.2.1. Entrega de contexto adecuado

Los LLM a menudo presentan alucinaciones, por lo que es esencial reducir la frecuencia de este fenómeno. Para lograrlo, la provisión de contexto dentro del prompt no es simplemente precisa, sino que resulta absolutamente indispensable. De hecho, la entrega de contexto adecuado dentro del prompt ha demostrado ser una medida altamente efectiva para reducir las alucinaciones, logrando una disminución de hasta un 99.88 porciento [6].

Por consiguiente, la correcta entrega de contexto dentro del prompt desempeña un papel fundamental en la generación de respuestas precisas a las consultas. Esto se debe a que, ya sea que el contexto proporcionado sea correcto, incorrecto o incluso irrelevante, el modelo de lenguaje lo utilizará como base para generar sus respuestas.

En el contexto del proyecto, la generación de respuestas se basa por completo en la entrega de contexto dentro del prompt, lo que a veces puede dar lugar a la transmisión de más información de la necesaria debido al funcionamiento del framework de Langchain. Esto puede llevar a situaciones en las que el modelo, influenciado por la información incorrecta o adicional proporcionada, genere respuestas que no reflejan un output con una respuesta en su totalidad correcta.

4.2.2. Limitaciones de la similitud de cosenos

La similitud del coseno, siendo este método más usado en modelos RAG para la extracción de contexto en la base de datos, como medida de similitud semántica en los embeddings, particularmente para palabras de alta frecuencia en tareas de procesamiento de lenguaje natural (NLP) como preguntas y respuestas (QA), recuperación de información (IR) y traducción automática (MT) presenta limitaciones en su uso, esto principalmente sucede debido a que la frecuencia de las palabras en los datos de entrenamiento afecta la geometría representacional de los embeddings contextualizados, siendo las palabras de baja frecuencia más concentradas geométricamente [20].

Por lo tanto, este problema se extrapola a que, en el momento de querer recuperar contexto pertinente de la base de datos, cuando se realiza el proceso de semejanza semántica entre el prompt y los vectores de la base de datos, este pueda recibir información no relacionada con el prompt, por lo que se enviar como contexto y puede dar oportunidad a alucinaciones.

4.2.3. Alucinaciones

El termino alucinación se refiere a la generación de textos o respuestas que exhiben corrección gramatical, fluidez y autenticidad, pero se desvían de las entradas de fuente proporcionadas (fidelidad) o no se alinean con la precisión factual (factualidad) [19]. Siendo en palabras más simples la entrega de información invetada por el modelo.

Dicho lo anterior, podemos decir que este es un gran factor de riesgo para el uso de una aplicación, porque a pesar de estar usando un sistema RAG, que dificulta la posibilidad de generar alucinaciones, sigue estando la posibilidad de que estas sucedan lo que puede entregar un output con información errónea y si es que no se revisa con criterio, se podría a llevar a cometer graves errores debido al uso de información que es directamente falsa.

4.2.4. Aprendizaje por Refuerzo con Retroalimentación Humana (RLHF)

Los modelos grandes de lenguaje suelen dar respuestas que a veces tanto política como moralmente no son correctas, por lo que las empresas tienen por objetivo alinear los valores humanos con los sistemas de aprendizaje automático y dirigir los algoritmos de aprendizaje hacia los objetivos e intereses de los humanos [21]. A esto se le llama Aprendizaje por Refuerzo con Retroalimentación Humana (RLHF), esto puede llegar a ser un problema si es que se busca realizar una aplicación en un usuario con una cultura diferente al proveedor del modelo o que si el usuario al no expresarse bien el modelo se confunda y no entregue una respuesta satisfactoria.

5 | Conclusiones

Bibliografía

[1] *¿Qué es la IA generativa y cuáles son sus aplicaciones? | Google Cloud*. URL: https://cloud.google.com/use-cases/generative-ai?hl=es.

[2] Syed Muhammad Fawad Ali y Robert Wrembel. "From conceptual design to performance optimization of ETL workflows: current state of research and open problems". En: *VLDB Journal* 26 (6 dic. de 2017), págs. 777-801. ISSN: 0949877X. DOI: 10.1007/S00778-017-0477-2/FIGURES/18. URL: https://link.springer.com/article/10.1007/s00778-017-0477-2.

[3] *Buscador Ambiental*. URL: https://www.buscadorambiental.cl/buscador/#/.

[4] Yi Cao y Jia Zhai. "Bridging the gap–the impact of ChatGPT on financial research". En: *Journal of Chinese Economic and Business Studies* 21 (2 2023), págs. 177-191. ISSN: 14765292. DOI: 10.1080/14765284.2023.2212434. URL: https://www.tandfonline.com/action/journalInformation?journalCode=rcea20.

[5] Armanda Cetrulo y Alessandro Nuvolari. "Industry 4.0: revolution or hype? Reassessing recent technological trends and their impact on labour". En: *Journal of Industrial and Business Economics* 46 (3 sep. de 2019), págs. 391-402. ISSN: 19724977. DOI: 10.1007/S40812-019-00132-Y/TABLES/1. URL: https://link.springer.com/article/10.1007/s40812-019-00132-y.

[6] Philip Feldman, James R. Foulds y Shimei Pan. "Trapping LLM Hallucinations Using Tagged Context Prompts". En: (jun. de 2023). URL: https://arxiv.org/abs/2306.06085v1.

[7] Yue Guo, Yi Yang y Ahmed Abbasi. "Auto-Debias: Debiasing Masked Language Models with Automated Biased Prompts". En: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. por Smaranda Muresan, Preslav Nakov y Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, mayo de 2022, págs. 1012-1023. DOI: 10.18653/v1/2022.acl-long.72. URL: https://aclanthology.org/2022.acl-long.72.

[8] Lakera. *The List of 11 Most Popular Open Source LLMs of 2023 | Lakera – Protecting AI teams that disrupt the world.* URL: https://www.lakera.ai/blog/open-source-llms.

[9] Patrick Lewis y col. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". En: *Advances in Neural Information Processing Systems* 33 (2020), págs. 9459-9474. URL: https://github.com/huggingface/transformers/blob/master/.

[10] *Ley Chile - Ley 20600 - Biblioteca del Congreso Nacional*. URL: https://www.bcn.cl/leychile/navegar?idNorma=1041361&idParte=9269911.

[11] Shreekant Mandvikar. "Factors to Consider When Selecting a Large Language Model: A Comparative Analysis". En: *International Journal of Intelligent Automation and Computing* 6 (3 ago. de 2023), págs. 37-40. URL: https://research.tensorgate.org/index.php/IJIAC/article/view/53.

[12] Puranjay Savar Mattas. "ChatGPT: A Study of AI Language Processing and its Implications". En: *International Journal of Research Publication and Reviews* 04 (02 2023), págs. 435-440. DOI: 10.55248/GENGPI.2023.4218.

[13] Mrwhosetheboss. *Can AI really save the World? ft. Bill Gates - YouTube*. URL: https://www.youtube.com/watch?v=l9m3IKG8i88&t=16s.

[14] N. K. Nagwani. "Summarizing large text collection using topic modeling and clustering based on MapReduce framework". En: *Journal of Big Data* 2 (1 dic. de 2015), págs. 1-18. ISSN: 21961115. DOI: 10.1186/S40537-015-0020-5/FIGURES/15. URL: https://link.springer.com/articles/10.1186/s40537-015-0020-5%20https://link.springer.com/article/10.1186/s40537-015-0020-5.

[15] OpenAI. *Embeddings - OpenAI API*. URL: https://platform.openai.com/docs/guides/embeddings/what-are-embeddings.

[16] OpenAI. *Introducing GPTs*. URL: https://openai.com/blog/introducing-gpts.

[17] OpenAI. *New models and developer products announced at DevDay*. URL: https://openai.com/blog/new-models-and-developer-products-announced-at-devday.

[18] Mirela Șorecău y Emil Șorecău. "AN ALTERNATIVE APPLICATION TO CHATGPT THAT USES RELIABLE SOURCES TO ENHANCE THE LEARNING PROCESS". En: XXIX (2023), pág. 2023. DOI: 10.2478/kbo-2023-0084.

[19] Hongbin Ye y col. "Cognitive Mirage: A Review of Hallucinations in Large Language Models". En: (sep. de 2023). URL: https://arxiv.org/abs/2309.06794v1.

[20] Kaitlyn Zhou y col. "Problems with Cosine as a Measure of Embedding Similarity for High Frequency Words". En: *Proceedings of the Annual Meeting of the Association for Computational Linguistics* 2 (mayo de 2022), págs. 401-423. ISSN: 0736587X. DOI: 10.18653/v1/2022.acl-short.45. URL: https://arxiv.org/abs/2205.05092v1.

[21] Banghua Zhu, Jiantao Jiao y Michael I. Jordan. "Principled Reinforcement Learning with Human Feedback from Pairwise or K-wise Comparisons". En: (ene. de 2023). ISSN: 26403498. URL: https://arxiv.org/abs/2301.11270v4.