**Instituto Tecnológico**
**y de Estudios Superiores de Occidente**
Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática
**Maestría en Sistemas Computacionales**


# OPTIMIZACIÓN DE PROMPTS EN MODELOS DE LENGUAJE USANDO ESTRUCTURAS SEMÁNTICAS

---

**TRABAJO RECEPCIONAL** que para obtener el **GRADO** de
**MAESTRO EN SISTEMAS COMPUTACIONALES**

Presenta: **ING. RODOLFO LUTHE NARVÁEZ**

Asesor **DR. LUIS MIGUEL ESCOBAR VEGA**

Tlaquepaque, Jalisco. julio de 2025.

# AGRADECIMIENTOS

El autor desea dar las gracias al ITESO, por la ayuda de todos los docentes y directivos que apoyaron con la elaboración de este documento, con docencia directa o con consejos para su elaboración. También a Oracle Inc. que apoyó con recursos financieros que permitieron la elaboración de este documento con mínimo estrés pecuniario.

# DEDICATORIA

El autor dedica esta tesis a su familia, sus profesores y amigos que apoyaron el programa de maestría que resultó en este trabajo, sin los cuales no hubiera sido posible.

# RESUMEN

Se presenta una breve introducción al enriquecimiento de prompts, el cual tiene como objetivo principal mejorar el rendimiento de LLMs con tareas de un usuario comercial, resolviendo de manera particular las alucinaciones y las respuestas incompletas que no identifican correctamente todas las entidades en la tarea. El trabajo enriquece los prompts originales de un usuario con una ontología OWL de cuatro maneras diferentes. Finalmente, se observa que no hubo ninguna mejora en las métricas de evaluación comparando el prompt original con el enriquecido.

# TABLA DE CONTENIDO

# LISTA DE FIGURAS

# LISTA DE TABLAS

# LISTA DE ACRÓNIMOS Y ABREVIATURAS

| | |
|---|---|
| LLM | Large language model |
| OWL | Ontology Web Language |

# 1. INTRODUCCIÓN

.

## 1.1. Antecedentes

Mejorar la tecnología de trasfondo de los LLMs es muy costoso y lento, evidenciado por los avances más recientes en LLMs, que consisten en hacer más grande la red de transformadores del modelo y no en cambios cualitativos de la tecnología [1], por lo que se han buscado maneras alternativas de minimizar los errores y maximizar la utilidad de los modelos. La búsqueda de mejoras en el rendimiento de los LLMs da origen a la disciplina de la Ingeniería de prompts [2], que aplica técnicas semánticas para disminuir el fenómeno de la alucinación en los LLMs. Si adecuamos los prompts podría mejorar el rendimiento de un LLM sin que este tenga que pasar por mejoras técnicas, cosa que hace muy atractivo el dedicar más recursos a las optimizaciones de prompts.

Se busca abordar la problemática actual de cómo optimizar las preguntas o inputs en un modelo generativo para obtener respuestas que se alineen más estrechamente con la información utilizada durante el entrenamiento del modelo de lenguaje. Estudios previos [3], [4] han demostrado que la calidad de las respuestas generadas por el modelo depende significativamente de la pregunta de entrada; en particular, cuanto más contexto se proporcione, mejor será el resultado. Sin embargo, existe una falta de información detallada sobre las estrategias más efectivas para enriquecer o aumentar el contexto de manera que el modelo pueda generar respuestas con mayor precisión y certeza.

El objetivo general de este trabajo es diseñar un sistema automatizado que traduzca prompts en lenguaje natural a ontologías OWL, utilizando estas para enriquecer entradas a LLMs

Se eligieron las ontologías porque son la manera que más sentido tiene si el problema es la falta de estructura y claridad en un prompt. El uso de una ontología hace explícitas las cosas que un LLM tendría que deducir por contexto y con eso minimiza las oportunidades de que se dé un error ocasionado porque el modelo se confundió con el prompt. Esta estructura facilitará agregar al prompt del usuario un contexto añadido, y este contexto se puede poner en el mismo formato de ontología que el prompt, quitando el impedimento al LLM de tener que comprender dos estructuras, una de lenguaje natural y otra más estructurada.

Existen otras alternativas que logran propósitos similares, como RAG o inclusión de ejemplos en el prompt. En este proyecto se trabajará con las ontologías porque RAG, mientras que puede reducir alucinaciones cuando tiene un corpus de conocimiento suficientemente extenso, no valida la coherencia lógica entre los conceptos presentes en la instrucción y las relaciones entre ellos. En el caso de la inclusión de ejemplos no se utilizaron dado que tienen la misma falta de validación de coherencia de conceptos y relaciones.

La utilización de estructuras ontológicas introduce ciertas desventajas y limitaciones a este proyecto, principalmente el rendimiento fuera de dominios con ontologías disponibles. La solución será especialmente efectiva cuando se pueda hacer uso de una ontología formal preexistente, o en dominios donde se pueda construir una, como en la ingeniería, medicina o derecho, pero en dominios donde no hay una ontología el sistema intentará construir una ontología de manera automática, pero no será tan fiable como en dominios donde ya existen.

En vista a la abundancia de evidencia de los errores que existen en los LLMs y la capacidad única de la optimización de prompts para aminorarlos este proyecto buscará hacer una herramienta que use las ontologías para mejorar los prompts de sus usuarios.

## 1.2. Justificación

Se busca abordar la problemática actual de cómo optimizar las preguntas o inputs en un modelo generativo para obtener respuestas que se alineen más estrechamente con la información utilizada durante el entrenamiento del modelo de lenguaje. Estudios previos [3], [4] han demostrado que la calidad de las respuestas generadas por el modelo depende significativamente de la pregunta de entrada; en particular, cuanto más contexto se proporcione, mejor será el resultado. Sin embargo, existe una falta de información detallada sobre las estrategias más efectivas para enriquecer o aumentar el contexto de manera que el modelo pueda generar respuestas con mayor precisión y certeza.

## 1.3. Problema

Mejorar la tecnología de trasfondo de los LLMs es muy costoso y lento, evidenciado por los avances más recientes en LLMs, que consisten en hacer más grande la red de transformadores del modelo y no en cambios cualitativos de la tecnología [1], por lo que se han buscado maneras alternativas de minimizar los errores y maximizar la utilidad de los modelos. La búsqueda de mejoras en el rendimiento de los LLMs da origen a la disciplina de la Ingeniería de prompts [2], que aplica técnicas semánticas para disminuir el fenómeno de la alucinación en los LLMs. Si adecuamos los prompts podría mejorar el rendimiento de un LLM sin que este tenga que pasar por mejoras técnicas, cosa que hace muy atractivo el dedicar más recursos a las optimizaciones de prompts.

## 1.4. Hipótesis

El enriquecimiento de los prompts con estructuras ontológicas resultará en prompts de mayor calidad que le serán más útiles a usuarios que un prompt sin enriquecer.

## 1.5. Objetivos

### *1.5.1. Objetivo General:*

Diseñar un sistema automatizado que traduzca prompts en lenguaje natural a ontologías OWL, utilizando estas para enriquecer entradas a LLMs.

El sistema propuesto aborda dos problemas principales en el uso de LLMs:

- Reducción de alucinaciones y errores semánticos en respuestas especializadas:
    - Los LLMs, aunque potentes, tienden a generar respuestas incorrectas o “alucinaciones” cuando se enfrentan a temas técnicos o especializados, ya que su conocimiento es estadístico y no siempre está alineado con la realidad factual o las relaciones formales

del dominio. Al convertir el prompt inicial en una ontología OWL, el sistema formaliza y estructura el conocimiento relevante antes de enviarlo al modelo, lo que ayuda a limitar el espacio de respuestas posibles y a guiar al LLM hacia resultados más precisos y verificables.

- Validación y control sobre la interpretación del prompt:
    - La ontología permite explicitar y validar los conceptos, relaciones y restricciones relevantes del dominio antes de que el modelo genere una respuesta, lo que reduce ambigüedades y mejora la trazabilidad del razonamiento del sistema.

Delimitación del alcance:

- El agente funciona preferentemente en dominios con ontologías disponibles:
    - La solución es especialmente efectiva en áreas donde existen ontologías formales o es viable construirlas (medicina, derecho, ingeniería, etc.), ya que la calidad y utilidad del sistema dependen de la cobertura y precisión de la ontología generada o utilizada.
- Aplicación fuera de dominios en los que existen ontologías formales:
    - En dominios sin ontologías preexistentes, el sistema puede intentar construirlas automáticamente, pero su rendimiento y fiabilidad serán menores, ya que la construcción automática de ontologías sigue siendo un reto abierto y requiere validación adicional por expertos.
- Tipo de preguntas:
    - El enfoque es más eficaz para preguntas factuales, de consulta estructurada o que requieren precisión semántica y lógica, y menos para tareas creativas o abiertas.

### *1.5.2. Objetivos Específicos:*

1. Desarrollar un pipeline de procesamiento lingüístico para extraer entidades, relaciones y restricciones semánticas de prompts.
2. Implementar un módulo de conversión a OWL que formalice la estructura jerárquica y lógica del prompt.
3. Validar la eficacia del prompt ontológico mediante métricas de rendimiento comparativas (con/sin ontología).

## 1.6. Novedad científica, tecnológica o aportación

- **Técnica:** Un protocolo estandarizado para convertir prompts en ontologías, compatible con herramientas como Protégé [5] y lenguajes de consulta SPARQL.
- **Teórica:** Evidencia empírica sobre cómo la estructuración ontológica mejora la alineación entre la intención del usuario y la salida del LLM.
- **Práctica:** Plantillas reutilizables de ontologías para dominios frecuentes (ej. asistencia médica, soporte técnico), reduciendo el tiempo de diseño de prompts.

# 2. ESTADO DEL ARTE O DE LA TÉCNICA

## 2.1. Trabajos que usan ontologías

El trabajo más relacionado a esta propuesta es OntoChatGPT [6], que investigó el uso de una ontología como parte de la ingeniería de prompt, definiendo en la ontología reglas para las respuestas que genera el LLM, dándole mayor capacidad de detectar entidades en un texto y contestar preguntas sobre ellas en base a bases de conocimiento externas.

También ha habido otros trabajos que usan las ontologías como una base de datos en el que el modelo puede buscar información como en el caso de Allemang et al [7], la manera más usual es usando consultas de SPARQL [8], que son usadas como el método por excelencia para extraer información de una red de conocimiento formada por ontologías.

Otro trabajo que busca mezclar el prompt con ontologías es el de Ronzano et al [9] que hizo uso de una ontología con conocimiento biomédico en un embedding de ChatGPT 3.5 para aumentar su conocimiento de dominio en esa área particular.

## 2.2. Trabajos con otras técnicas

Existen intentos de hacer fine-tuning de LLMs con ontologías, como el caso de Baldazzi [10] que se separa del resto de los trabajos mencionados porque no está tratando de hacer RAG o algún tipo de ingeniería de prompt, si no que está generando un corpus para ajustar la conducta del LLM para el área de dominio de las ontologías que le provee

Existen también proyectos como el de Yang et al [11] que no usan ontologías formales, pero hacen uso de un grafo de conocimiento que sirve el mismo propósito y tiene una estructura muy similar a una ontología

Todos esto proyectos tienen el factor común de usar redes semánticas o grafos para ayudarle al LLM a recordar hechos, detalles de entidades y las relaciones entre conceptos y las mismas entidades, variando solamente en el alcance del grafo asistente, en el caso de OntoChatGPT hacen consultas de SPARQL para generar el grafo que le ayudará al LLM, mientras que muchos de los demás incluyeron un grafo semántico estático con conocimiento precargado.

Una diferencia de OntoChatGPT con lo que se quiere para este proyecto es que hacen uso de un ingeniero de datos para que arme el prompt final usando herramientas que se desarrollaron en el pasado. Usan su propio código para generar las consultas de SPARQL y no nos han dado acceso a ese para poderlo usar en este trabajo. También existen trabajos que transforman texto natural en consultas de SPARQL como FREyA y PAROT [12], [13], [14], pero no vienen en una librería de Python pública que se pueda usar para este trabajo, así que en este trabajo usaremos el LLM para generar las consultas en vez de recrear un módulo de procesamiento de texto.

# 3. MARCO TEÓRICO/CONCEPTUAL

## 3.1. LLMs

Son un tipo de modelo caracterizado por su habilidad de procesar y generar lenguaje natural humano para fines generalizados, contrastando con la mayoría de los modelos de Machine Learning (ML) que son diseñados para un solo fin específico y no suelen funcionar correctamente fuera de esos límites definidos por su caso de uso.

Esta capacidad se obtiene entrenando al modelo con volúmenes gigantescos de datos, consumiendo miles de millones de parámetros al entrenarse y consumiendo cantidades proporcionales de recursos de cómputo al hacerlo.

Los LLMs suelen componerse de capas de redes neuronales de tipo transformador [15], que es el descubrimiento tecnológico que permitió el uso a escala de los LLMs a escala de producción. Este tipo de modelo parece capaz de extraer estructuras semánticas como las ontologías de su entrenamiento.

Ejemplos de este tipo de modelos son: chatGPT [16], Bard [17], Claude [18] y Llama2 [19]

## 3.2. Ingeniería de Prompts

Esta rama de la ingeniería se preocupa principalmente por hacer más eficiente los LLMs no mediante mejoras técnicas a la tecnología subyacente o los algoritmos que emplean, si no que dedica sus esfuerzos a identificar las técnicas de redacción de prompts que mejores resultados obtienen cuando se usan en conjunto con un LLM [6], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29].

Un prompt constituye el texto en lenguaje natural que describe la tarea que una IA debe realizar, pudiendo ser una consulta, un comando o una declaración más larga que incluya contexto, instrucciones e historial de conversación [30], La evolución histórica de la ingeniería de prompts se remonta a 2018, cuando los investigadores propusieron por primera vez que todas las tareas previamente separadas en el procesamiento de lenguaje natural (NLP) podrían formularse como un problema de respuesta a preguntas sobre un contexto [31].

Una de las razones por las que existe esta área de investigación es una de las fallas más comunes de los LLMs, que es la llamada “hallucination” [32], [33], entendida como una mentira que el LLM cree que es real, aunque no se vea reflejada en el mundo real o en los datos de entrenamiento. Este tipo de errores reducen la utilidad de los modelos, así que cualquier mejora no solo en este aspecto es de vital importancia.

Existen servicios que optimizan los prompts [34] y guías paso a paso para tomar mejores decisiones al redactar dirigidas a un público sin educación técnica [35], [36], inclusive hay cursos apoyados por los creadores de ciertos LLMs [37] que tratan de capacitar a los potenciales usuarios de LLMs sobre redacción de prompts.

### 3.2.1. Técnicas Fundamentales

Hay una variedad de técnicas de ingeniería de prompts que abarcan un espectro amplio de metodologías, desde métodos fundamentales como el prompting de cero disparos (zero-shot) y pocos disparos (few-shot)

hasta enfoques más intrincados como el prompting de "cadena de código" [38], pero de particular interés son las aplicaciones de ontologías vistas en [6] donde se compararon diferentes maneras de optimizar prompts y se utilizó una basada en ontologías.

Una técnica particularmente relevante es el prompting de cadena de pensamiento (chain-of-thought), que descompone una pregunta compleja en partes lógicas más pequeñas que imitan un tren de pensamiento, ayudando al modelo a resolver problemas en una serie de pasos intermedios en lugar de responder directamente la pregunta [39]. Esta técnica mejora significativamente la capacidad de razonamiento del modelo al proporcionar un camino estructurado hacia la solución.

Otros trabajos han encontrado que ciertas técnicas pueden mejorar el rendimiento de un LLM, aunque sean tan sencillas como incluir una sola frase en el prompt [40]. Otras técnicas pueden ser más sofisticadas, como un algoritmo que busque documentos relacionados con el prompt antes de que el LLM lo procese solo para aumentar el contexto con que hace el procesamiento final. La técnica que se escogió para este trabajo es la de transformar el prompt base en lenguaje natural a una ontología.

## 3.3. Ontologías

“La ontología es la rama de la filosofía que se dedica a reflexionar sobre los modos esenciales de existencia de las cosas” [41] pero en términos prácticos podemos decir que una ontología es una manera de organizar información con una estructura consistente (ejemplos familiares a las ciencias de la computación pueden ser formatos de archivo). Esto implica descomponer el lenguaje natural y amigable en un formato consistente que extraiga los elementos del mensaje como los objetos, sus propiedades y las relaciones que tienen entre ellos.

Ejemplos de ontologías pueden verse en estándares como OWL functional syntax [29] que codifican los elementos mencionados en un estándar.

## 3.4. OWL

El Web Ontology Language (OWL) constituye una familia de lenguajes de representación del conocimiento para la autoría de ontologías. Las ontologías representan una forma formal de describir taxonomías y redes de clasificación, definiendo esencialmente la estructura del conocimiento para varios dominios: los sustantivos que representan clases de objetos y los verbos que representan relaciones entre los objetos. A diferencia de las jerarquías de clases en la programación orientada a objetos, las ontologías están diseñadas para representar información en Internet [42].

OWL se caracteriza por su semántica formal y está construido sobre el estándar del World Wide Web Consortium (W3C) para objetos llamado Resource Description Framework (RDF) [43]. El lenguaje proporciona una rica colección de operadores para formar descripciones de conceptos y ha sido diseñado para ser compatible con los estándares web existentes. Esta compatibilidad es crucial para el desarrollo de la Web Semántica, donde los recursos deben ser más accesibles a los procesos automatizados mediante anotaciones de metadatos semánticos.

## 3.5. Marco Conceptual de Integración Ontología-LLM

La integración de ontologías con LLMs representa un paradigma emergente que combina la representación formal del conocimiento con las capacidades generativas de los modelos de lenguaje [44]. Un marco novedoso de ajuste de prompts impulsado por ontologías emplea razonamiento basado en conocimiento para refinar y expandir los prompts del usuario con razonamiento contextual de la tarea y descripciones del estado del entorno basadas en conocimiento. Esta integración asegura que el conocimiento específico del dominio se incorpore al prompt, garantizando planes de tareas semánticamente precisos y conscientes del contexto.

El marco teórico subyacente demuestra que los modelos transformadores, cuando se proporcionan con prompts cuidadosamente diseñados, pueden actuar como un sistema computacional configurable emulando una red neuronal "virtual" durante la inferencia. Los prompts de entrada se traducen efectivamente en la configuración de red correspondiente, permitiendo a los LLMs ajustar sus cálculos internos dinámicamente [45]. Esta construcción establece una teoría de aproximación para funciones diferenciables, demostrando que los transformadores pueden aproximar tales funciones con precisión arbitraria cuando son guiados por prompts estructurados apropiadamente.

La utilización de ontologías en la ingeniería de prompts ofrece múltiples ventajas significativas. Primero, las ontologías proporcionan una representación estructurada y formal del conocimiento del dominio, lo que permite una comprensión más precisa del contexto y las relaciones entre conceptos. Segundo, la representación ontológica facilita el razonamiento automático sobre el conocimiento, permitiendo que el sistema inferir relaciones implícitas y mantenga consistencia semántica. Tercero, la formalización ontológica permite la validación automática de la coherencia y completitud del conocimiento representado, reduciendo errores semánticos en la generación de planes simbólicos

## 3.6. Metodologías de Evaluación para LLMs

### 3.6.1. Métricas Fundamentales de Evaluación

La evaluación de los LLMs requiere un enfoque integral que emplee una gama de medidas para evaluar varios aspectos de su rendimiento [46]. Las métricas de precisión y rendimiento constituyen el núcleo de la evaluación de LLMs. La perplejidad es una métrica fundamental que mide la capacidad de un LLM para predecir la siguiente palabra en una secuencia, calculándose a través de la probabilidad, probabilidad inversa y normalización. Puntuaciones de perplejidad más bajas indican que el modelo predice la siguiente palabra con mayor precisión, reflejando un mejor rendimiento.

La precisión representa la proporción de predicciones correctas realizadas por el modelo y es ampliamente utilizada para tareas de clasificación. Sin embargo, en el contexto de tareas de generación de extremo abierto, la precisión puede ser engañosa, ya que la "corrección" de la salida no es tan directa de definir como en tareas como análisis de sentimientos o clasificación de temas. Por lo tanto, la precisión debe complementarse con otras métricas al evaluar LLMs para tareas generativas complejas.

### 3.6.2. Metodologías de Evaluación Integral

Las metodologías de evaluación robustas para LLMs integran enfoques tanto cuantitativos como cualitativos. Los conjuntos de datos de referencia proporcionan tareas estandarizadas que permiten el análisis comparativo entre diferentes modelos, incluyendo GLUE (General Language Understanding Evaluation), SuperGLUE y SQuAD (Stanford Question Answering Dataset). Estos benchmarks establecen una línea base para el rendimiento del modelo y facilitan la comparación sistemática.

La evaluación humana permanece como un estándar de oro para evaluar los aspectos matizados de las salidas de LLM que las métricas automatizadas podrían pasar por alto. Los métodos de evaluación directa involucran la recopilación de retroalimentación de jueces humanos utilizando encuestas y escalas de calificación, capturando aspectos cualitativos de la calidad del texto como fluidez, coherencia y relevancia. El juicio comparativo implica técnicas como la comparación por pares, donde los evaluadores humanos comparan directamente las salidas de diferentes modelos, proporcionando una clasificación relativa del rendimiento del modelo.

# 4. DESARROLLO METODOLÓGICO

## 4.1. Levantamiento de requerimientos

El proceso por el que pasa un prompt del usuario consiste en tres etapas generales:

1. Prompt del usuario

Este es el texto que genera el usuario, en lenguaje natural. Contiene las instrucciones que el usuario quiere que el LLM realice. Este prompt se combina con el prompt base para generar un tercer prompt que le dice al LLM que ahora queremos crear un prompt basado en el del usuario con una ontología. A esto nos referiremos como la petición.

2. Prompt en ontología

Es un texto en el formato de la ontología que contiene las instrucciones especificadas en el prompt inicial del usuario. Contendrá las entidades y las relaciones entre ellas en un texto que se pueda usar con el mismo LLM que lo generó.

3. Respuesta final

Es el resultado de usar el prompt generado en el paso 2 con el LLM, ocasionando que el LLM siga las instrucciones que el usuario escribió en el paso 1, pero dentro de la estructura ontológica junto con su contexto expandido.

El proceso lo podemos ver ilustrado en la imagen 1:

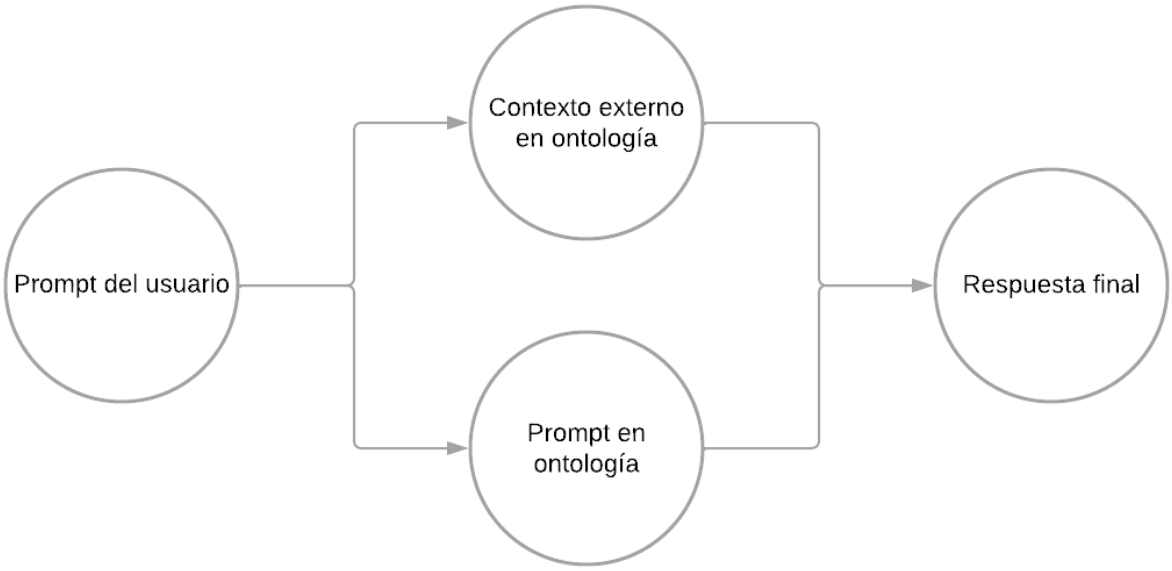


**Figura 1 Etapas del proceso de transformación del prompt**

Podemos ver la arquitectura que lleva a cabo esta transformación en la Imagen 2, que nos muestra las partes componentes del sistema y la manera en la que se pasan información entre sí.

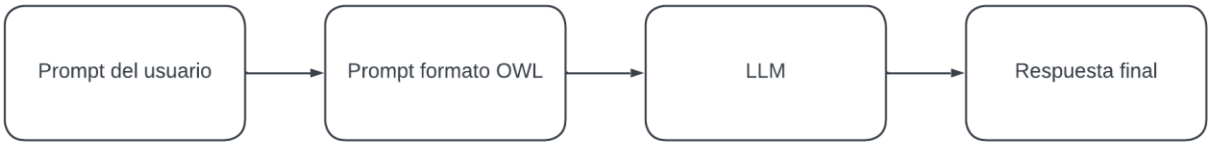


**Figura 2 Arquitectura del programa y etapas por las que pasa el prompt del usuario.**

Esta arquitectura tiene un solo elemento de entrada, el prompt del usuario y una sola salida que es el resultado del prompt agregado a una ontología pasando por un modelo.

Los componentes internos son:

1. LLM
    a. Ejemplos: Bard, Llama 2, Cohere, etc…
2. Sistema
    a. Interactúa con el LLM; lo alimenta con el prompt en ontología.
    b. Modifica el prompt con técnicas de procesamiento natural del lenguaje con el propósito de extraer las entidades del prompt y describir sus relaciones en una ontología OWL

Podemos ver un diagrama del sistema en la Imagen 3, incluyendo la metodología de evaluación

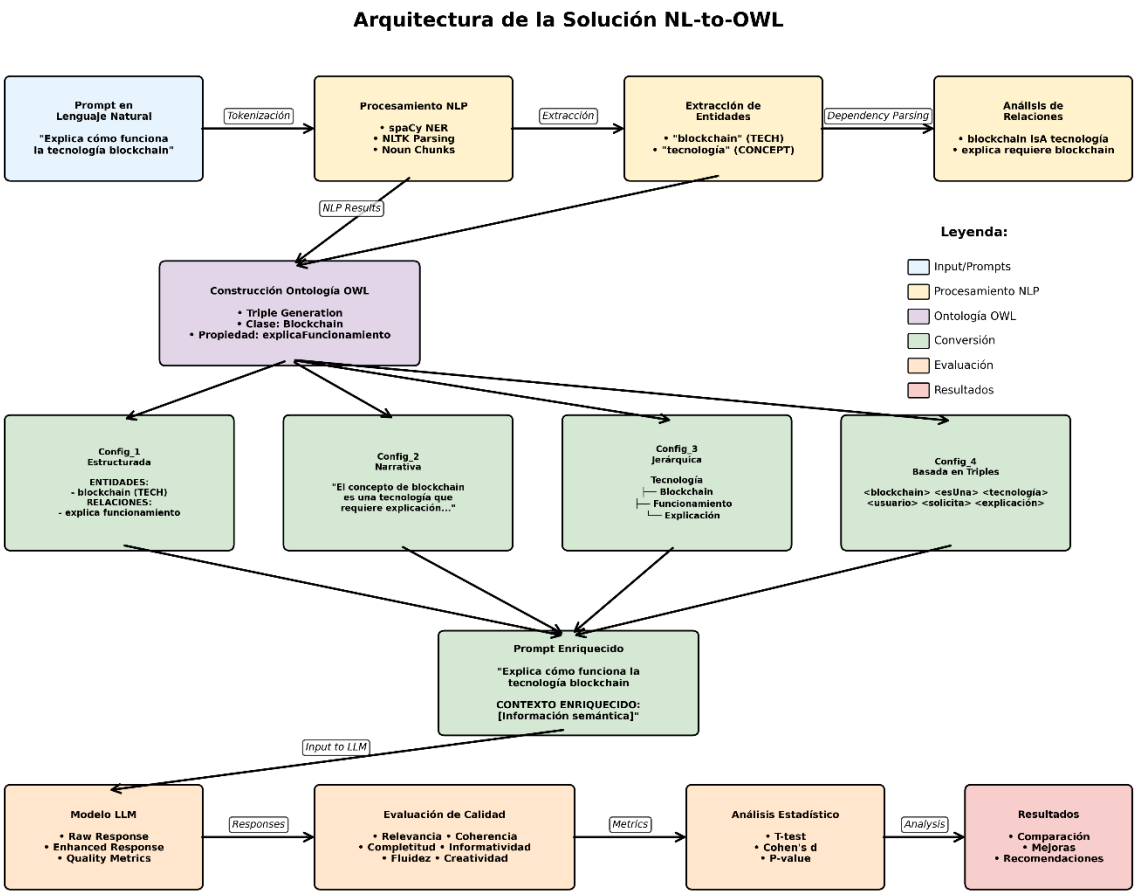


**Figura 3 Etapas detalladas por las que pasa el prompt del usuario en la solución.**

La explicación de las fases por las que pasa el prompt:

### Fase 1: Captura del Prompt Original

**El sistema inicia con un prompt en lenguaje natural tal como lo escribiría un usuario real:**

**Ejemplo: "Explica cómo funciona la tecnología blockchain"**

Usaremos el mismo ejemplo en el resto de las fases para poder apreciar mejor as transformaciones aplicadas por este sistema.

### Fase 2: Procesamiento de Lenguaje Natural

El prompt entra al módulo procesador que realiza múltiples análisis simultáneos con las siguientes técnicas

- Reconocimiento de Entidades Nombradas (NER): Identifica "blockchain" como TECNOLOGÍA
- Análisis de dependencias sintácticas: Mapea relaciones gramaticales

- Extracción de noun chunks: Identifica "tecnología blockchain" como concepto compuesto
- Normalización de texto: Convierte a formas canónicas para ontologías

**Fase 3: Extracción de Entidades Semánticas**

El sistema identifica entidades clave y las clasifica:

```
# Entidades extraídas del ejemplo
entities = [
    Entity(text="blockchain", label="TECHNOLOGY", normalized="blockchain"),
    Entity(text="tecnología", label="CONCEPT", normalized="tecnologia"),
    Entity(text="funcionamiento", label="PROCESS", normalized="funcionamiento")
]
```

**Fase 4: Análisis de Relaciones**

Utilizando análisis de dependencias, el sistema identifica relaciones semánticas:

```
relationships = [
    Relationship(source="blockchain", predicate="isA", target="tecnología"),
    Relationship(source="usuario", predicate="solicita", target="explicación"),
    Relationship(source="explicación", predicate="describe", target="funcionamiento")
]
```

**Fase 5: Construcción de Ontología OWL**

El conversor NL-to-OWL transforma las entidades y relaciones en una ontología formal en su formato OWL completo.

**Fase 6: Estrategias de Conversión OWL-to-Text**

El sistema implementa cuatro estrategias diferentes para convertir la ontología de vuelta a texto enriquecido, las mostramos con ejemplos en pseudocódigo con el mismo prompt imaginario de blockchain:

- Estructurada

```
ENTIDADES:
- blockchain (TECHNOLOGY): distributed ledger, cryptographic hashing
- tecnología (CONCEPT): underlying system, computational method

RELACIONES:
- blockchain isA tecnología
- usuario solicita explicación
```

- Narrativa

```
"En el contexto de la consulta sobre blockchain, es importante entender que
se trata de una tecnología específica que implementa un sistema de registro
```

distribuido. La explicación del funcionamiento requiere comprender..."

- Jerárquica

```
Tecnología
├── Blockchain
    ├── Características
    │   ├── Descentralización
    │   └── Inmutabilidad
    └── Funcionamiento
        ├── Bloques
        └── Cadena
```

- Triplas

```
<blockchain> <esUna> <tecnología>
<blockchain> <tiene> <funcionamiento>
<usuario> <solicita> <explicación>
<explicación> <describe> <funcionamiento>
```

**Fase 7: Generación del Prompt Enriquecido**

Cada estrategia produce un prompt enriquecido que mantiene la intención original, pero añade contexto semántico, usamos la configuración narrativa para este ejemplo. Pero se usan las cuatro para las pruebas:

```
Prompt Original: "Explica cómo funciona la tecnología blockchain"
```

```
Prompt Enriquecido:
"Explica cómo funciona la tecnología blockchain

CONTEXTO SEMÁNTICO:
En el contexto de esta consulta, blockchain es una tecnología específica
que requiere explicación de su funcionamiento. Los conceptos clave incluyen
la naturaleza distribuida del sistema y los mecanismos criptográficos..."
```

Estos prompt son usados con un LLM y la respuesta es evaluada en la siguiente fase

**Fase 8: Evaluación Comparativa en LLM**

Se le pasan ambos prompts a un modelo de DialoGPT para ser evaluados

# 5. RESULTADOS Y DISCUSIÓN

## 5.1. Resultados

La evaluación experimental del sistema se diseñó como un estudio comparativo exhaustivo para determinar la efectividad del enriquecimiento semántico de prompts en la mejora de la calidad de respuestas de modelos de lenguaje grandes (LLMs).

El diseño experimental corresponde a un enfoque comparativo dentro de un mismo tema, utilizando un diseño factorial. Se contemplan como variables independientes cuatro estrategias distintas de conversión de OWL a texto: estructurada, narrativa, jerárquica y basada en triples. Las variables dependientes están representadas por seis métricas de calidad, las cuales incluyen:

1. Relevancia,
2. Coherencia,
3. Completitud,
4. Que tanta información provee,
5. Fluidez
6. Creatividad.

En cuanto a las categorías de los prompts, se consideran cinco dominios temáticos: creativo, analítico, técnico, factual y de resolución de problemas. El tamaño de la muestra se ha establecido en 25 prompts por configuración, lo que implica la utilización de cinco prompts para cada una de las cinco categorías mencionadas.

**Rendimiento por Configuración**

Los resultados experimentales revelan que el enriquecimiento semántico OWL no produjo las mejoras esperadas en la calidad de respuestas de LLMs. La tabla 1 presenta el resumen de resultados para cada configuración:

**Tabla 1 Resultados por configuración**

| Configuración | Mejora General | Tamaño del Efecto | Valor p | Categorías Significativas |
|---|---|---|---|---|
| Estructurada (Config_1) | -3.33% | -0.295 | 0.100 | Ninguna |
| Narrativa (Config_2) | -3.33% | -0.295 | 0.100 | Ninguna |
| Jerárquica (Config_3) | -1.67% | -0.167 | 0.100 | Ninguna |
| Basada en Triplas (Config_4) | 0.00% | 0.000 | 0.100 | Ninguna |

El análisis de los hallazgos anteriores muestra que la mejora promedio general observada fue de -2.08%, lo que representa una ligera degradación del rendimiento respecto al prompt sin alterar. La mejor configuración identificada correspondió a la Config_4, basada en triples, la cual no mostró ni mejoras ni deterioros significativos. En el extremo contrario, las configuraciones que obtuvieron los peores resultados fueron la Config_1 y la Config_2, ambas con una degradación del -3.33%. Finalmente, es importante

destacar que ninguna de las configuraciones evaluadas arrojó mejoras que fueran estadísticamente significativas, considerando un umbral de $p < 0.05$.

**Tabla 2 Resultados por categoría de prompt.**

| Categoría | Mejora Promedio | Mejor Config | Peor Config |
|---|---|---|---|
| Creativo | -2.08% | Config_1, Config_3, Config_4 (0.00%) | Config_2 (-8.33%) |
| Analítico | -2.08% | Config_2, Config_3, Config_4 (0.00%) | Config_1 (-8.33%) |
| Técnico | 0.00% | Todas las configuraciones | Ninguna |
| Factual | 0.00% | Todas las configuraciones | Ninguna |
| Resolución de Problemas | -6.25% | Config_4 (0.00%) | Config_1, Config_2, Config_3 (-8.33%) |

En cuanto a las observaciones por categoría, se identificó que las categorías técnica y factual no presentaron un cambio, reflejado en un 0.00% en todas las configuraciones. Por otro lado, la categoría de resolución de problemas resultó ser la más impactad, con una disminución promedio del 6.25%. Finalmente, los prompts correspondientes a las categorías creativa y analítica evidenciaron una variabilidad significativa en función de la estrategia de configuración usada.

**Análisis Estadístico**

Los resultados de pruebas t-student mostraron para todas las configuraciones valores de $p = 0.100 > \alpha = 0.05$, lo cual nos lleva a la conclusión de que no existe evidencia estadística de mejora en ninguna configuración

La interpretación de tamaños de efecto cuenta una historia muy similar como podemos ver:

- Config_1 y Config_2: $d = -0.295$ (efecto pequeño negativo)
- Config_3: $d = -0.167$ (efecto trivial negativo)
- Config_4: $d = 0.000$ (sin efecto)

Conclusión: Los efectos observados son de magnitud pequeña a trivial, sugiriendo que las diferencias no son prácticamente significativas.

**Interpretación de Resultados**

## Hipótesis vs. Resultados Observados

Hipótesis inicial: El enriquecimiento semántico mediante ontologías OWL mejoraría la calidad de respuestas de LLMs al proporcionar contexto estructurado.

Resultados observados:

- **Hipótesis rechazada:** No se observaron mejoras significativas
- En algunos casos, se observó degradación del rendimiento
- Únicamente la estrategia basada en triples mantuvo neutralidad

## 5.2. Discusión

Una posible explicación de los resultados observados radica en las limitaciones propias del enriquecimiento semántico. En primer lugar, la sobrecarga informacional puede haber generado ruido adicional en el contexto, dificultando la interpretación por parte del modelo. Asimismo, la información ontológica proporcionada podría no ser relevante para todas las tareas, lo que limita su utilidad en ciertos escenarios. Además, el formato resultante de las estrategias de conversión OWL-to-text podría no ser lo suficientemente natural para los modelos de lenguaje, que están entrenados con este tipo de lenguaje, afectando así su capacidad de procesamiento óptimo.

Por otro lado, es importante considerar las características del modelo base. Los modelos de lenguaje de última generación ya poseen un conocimiento semántico implícito considerable, lo que puede hacer que el enriquecimiento adicional resulte redundante respecto a la información de su entrenamiento original. A esto se suma el posible sesgo de formato, ya que estos modelos suelen estar optimizados para procesar texto natural y no necesariamente estructuras textuales artificiales o altamente estructuradas.

Finalmente, deben tomarse en cuenta ciertas limitaciones metodológicas que podrían haber influido en los resultados. El tamaño de la muestra, limitado a 25 prompts por configuración, podría no ser suficiente para detectar diferencias significativas. Por otra parte, las métricas de evaluación automatizadas empleadas pueden no ser capaces de captar mejoras sutiles en la calidad de las respuestas. Además, el dominio de prueba seleccionado podría no estar adecuadamente alineado con los beneficios potenciales del enriquecimiento semántico, lo que restringe la observación de posibles ventajas.

# 6. CONCLUSIONES

## 6.1. Conclusiones

Los resultados obtenidos no permitieron confirmar la hipótesis principal, lejos de ello, parece que el enriquecimiento semántico mediante ontologías OWL no produjo ninguna mejora en la calidad de las respuestas generadas por los modelos. Por el contrario, se identificó una tendencia hacia la degradación del rendimiento, con una disminución promedio del 2.1%. La única estrategia que logró mantener un desempeño neutro, sin degradación, fue la basada en triples RDF.

El análisis evidenció que el impacto del enriquecimiento semántico varía según el tipo de tarea evaluada. Particularmente, las tareas de resolución de problemas resultaron ser las más susceptibles a la degradación del rendimiento. La creatividad fue la única dimensión que se vio afectada de manera significativa y negativa, mientras que otras dimensiones como la relevancia y la coherencia permanecieron neutrales.

Desde el punto de vista técnico y metodológico, se demostró la viabilidad de la arquitectura propuesta. El sistema desarrollado mostró la capacidad funcional para extraer entidades y relaciones para convertirlas en texto enriquecido de manera automatizada.

En términos teóricos, los resultados sugieren que los modelos de lenguaje contemporáneos ya cuentan con representaciones semánticas implícitas tan robustas que pueden limitar el impacto de un enriquecimiento externo en un formato altamente estructurado.

## 6.2. Trabajo Futuro

Los resultados de este análisis sugieren la necesidad de enfoques más sofisticados y adaptativos. A continuación, se presentan las principales ideas de investigación que emergen de este estudio

1. Desarrollo de estrategias de enriquecimiento adaptativas dependiendo del tipo de tarea asignada por el usuario
2. Integración con grafos de conocimiento existentes
3. Evaluación con modelos diferentes

# BIBLIOGRAFÍA


[1] "The history, timeline, and future of LLMs." Accessed: Feb. 16, 2024. [Online]. Available: https://toloka.ai/blog/history-of-llms/

[2] "7 Reasons Prompt Engineering is Essential for Organizations." Accessed: Feb. 02, 2024. [Online]. Available: https://inclusioncloud.com/insights/blog/prompt-engineering-organizations/

[3] Y. Chen, J. Arkin, Y. Hao, Y. Zhang, N. Roy, and C. Fan, "PRompt Optimization in Multi-Step Tasks (PROMST): Integrating Human Feedback and Preference Alignment," Feb. 2024, Accessed: Feb. 16, 2024. [Online]. Available: https://arxiv.org/abs/2402.08702v1

[4] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, "A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications," Feb. 2024, Accessed: Feb. 16, 2024. [Online]. Available: https://arxiv.org/abs/2402.07927v1

[5] "protégé." Accessed: Jul. 15, 2025. [Online]. Available: https://protege.stanford.edu/

[6] O. Palagin, V. Kaverinsky, A. Litvin, and K. Malakhov, "OntoChatGPT Information System: Ontology-Driven Structured Prompts for ChatGPT Meta-Learning," vol. 170, no. 2, 2023, doi: 10.47839/ijc.22.2.3086.

[7] D. Allemang and J. Sequeda, "Increasing the LLM Accuracy for Question Answering: Ontologies to the Rescue!," May 2024, Accessed: Sep. 03, 2024. [Online]. Available: https://arxiv.org/abs/2405.11706v1

[8] B. DuCharme, "Learning SPARQL Querying and Updating with SPARQL 1.1," *O'Reilly Media*, vol. 12, no. 1, pp. 1–10, 2010.

[9] F. Ronzano and J. Nanavati, "Towards Ontology-Enhanced Representation Learning for Large Language Models," May 2024, Accessed: Sep. 03, 2024. [Online]. Available: https://arxiv.org/abs/2405.20527v1

[10] T. Baldazzi, L. Bellomarini, S. Ceri, A. Colombo, A. Gentili, and E. Sallinger, "Fine-tuning Large Enterprise Language Models via Ontological Reasoning," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 14244 LNCS, pp. 86–94, Jun. 2023, doi: 10.1007/978-3-031-45072-3_6.

[11] L. Yang, H. Chen, Z. Li, X. Ding, and X. Wu, "Give Us the Facts: Enhancing Large Language Models with Knowledge Graphs for Fact-aware Language Modeling," *IEEE Trans Knowl Data Eng*, vol. 36, no. 7, pp. 3091–3110, Jun. 2023, doi: 10.1109/TKDE.2024.3360454.

[12] "GitHub - onexpeters/PAROT: Natural Language to SPARQL tool." Accessed: Sep. 30, 2024. [Online]. Available: https://github.com/onexpeters/PAROT

[13] “GitHub - danicadamljanovic/freya: FREyA is a Natural Language Interface for Querying Ontologies.” Accessed: Sep. 30, 2024. [Online]. Available: https://github.com/danicadamljanovic/freya

[14] “GitHub - vasugr/Natural-Language-To-SPARQL: Translate a query in natural language to SPARQL query with an interactive web-based interface.” Accessed: Sep. 30, 2024. [Online]. Available: https://github.com/vasugr/Natural-Language-To-SPARQL

[15] R. Meerit, “What Is a Transformer Model? | NVIDIA Blogs.” Accessed: Nov. 27, 2023. [Online]. Available: https://blogs.nvidia.com/blog/what-is-a-transformer-model/

[16] OpenAI, “ChatGPT.” Accessed: Nov. 24, 2023. [Online]. Available: https://openai.com/chatgpt

[17] Google, “Bard - Chat Based AI Tool from Google, Powered by PaLM 2.” Accessed: Nov. 24, 2023. [Online]. Available: https://bard.google.com/

[18] Anthropic, “Product \ Anthropic.” Accessed: Nov. 26, 2023. [Online]. Available: https://www.anthropic.com/product

[19] Meta, “Introducing Code Llama, a state-of-the-art large language model for coding.” Accessed: Nov. 24, 2023. [Online]. Available: https://ai.meta.com/blog/code-llama-large-language-model-coding/

[20] K. Shum, S. Diao, T. Zhang, and T. Hong Kong, “Automatic Prompt Augmentation and Selection with Chain-of-Thought from Labeled Data,” Feb. 2023, Accessed: Nov. 26, 2023. [Online]. Available: https://arxiv.org/abs/2302.12822v1

[21] C. Zheng, Z. Liu, E. Xie, Z. Li, and Y. Li, “Progressive-Hint Prompting Improves Reasoning in Large Language Models,” Apr. 2023, Accessed: Nov. 26, 2023. [Online]. Available: https://arxiv.org/abs/2304.09797v5

[22] Z. Chen, K. Zhou, B. Zhang, Z. Gong, W. X. Zhao, and J.-R. Wen, “ChatCoT: Tool-Augmented Chain-of-Thought Reasoning on Chat-based Large Language Models,” May 2023, Accessed: Nov. 26, 2023. [Online]. Available: https://arxiv.org/abs/2305.14323v3

[23] S. Pitis, M. R. Zhang, A. Wang, and J. Ba, “Boosted Prompt Ensembles for Large Language Models,” Apr. 2023, Accessed: Nov. 26, 2023. [Online]. Available: https://arxiv.org/abs/2304.05970v1

[24] O. V. Palagin, K. S. Malahov, V. Yu. Velychko, and O. S. Shchurov, “Designing and program implementation of the subsystem for creation and use of the ontological knowledge base of the scientific employee publications,” *PROBLEMS IN PROGRAMMING*, vol. 0, no. 2, pp. 72–81, 2017, doi: 10.15407/PP2017.02.072.

[25] R. Pryzant, D. Iter, J. Li, Y. T. Lee, C. Zhu, and M. Zeng, “Automatic Prompt Optimization with ‘Gradient Descent’ and Beam Search,” May 2023, Accessed: Nov. 26, 2023. [Online]. Available: https://arxiv.org/abs/2305.03495v2

[26] A. Alcaraz, “Integrating Ontologies with Large Language Models for Decision-Making | by Anthony Alcaraz | Artificial Intelligence in Plain English.” Accessed: Nov. 24, 2023. [Online]. Available: https://ai.plainenglish.io/integrating-ontologies-with-large-language-models-for-decision-making-bb1c600ce5a3

[27] Y. Betterest Li and K. Wu, “SPELL: Semantic Prompt Evolution based on a LLM,” Oct. 2023, Accessed: Nov. 21, 2023. [Online]. Available: https://arxiv.org/abs/2310.01260v1

[28] C. Yang *et al.*, “Large Language Models as Optimizers,” Sep. 2023, Accessed: Nov. 21, 2023. [Online]. Available: https://arxiv.org/abs/2309.03409v1

[29] P. Mateiu and A. Groza, “Ontology engineering with Large Language Models,” 2023.

[30] D. Genkina, “AI Prompt Engineering is Dead: Long live AI prompt engineering,” *IEEE Spectr*, Mar. 2024, Accessed: Jun. 06, 2025. [Online]. Available: https://spectrum.ieee.org/prompt-engineering-is-dead

[31] B. McCann, N. S. Keskar, C. Xiong, and R. Socher, “The Natural Language Decathlon: Multitask Learning as Question Answering,” Jun. 2018, Accessed: Jun. 06, 2025. [Online]. Available: http://arxiv.org/abs/1806.08730

[32] M. I *et al.*, “Minimizing Factual Inconsistency and Hallucination in Large Language Models,” *Proceedings of ACM Web Conference 2024 (WWW â•Ž24 Companion)*, vol. 1, Nov. 2023, doi: XXXXXXX.XXXXXXX.

[33] H. Kang and X.-Y. Liu, “Deficiency of Large Language Models in Finance: An Empirical Examination of Hallucination,” Nov. 2023, Accessed: Nov. 26, 2023. [Online]. Available: https://arxiv.org/abs/2311.15548v1

[34] PrompPerfect, “PromptPerfect - Elevate Your Prompts to Perfection. Prompt Engineering, Optimizing, Debugging and Hosting.” Accessed: Nov. 26, 2023. [Online]. Available: https://promptperfect.jina.ai/

[35] “Optimizing your prompt.” Accessed: Nov. 26, 2023. [Online]. Available: https://docs.anthropic.com/claude/docs/optimizing-your-prompt

[36] G. Mileva, “The Ultimate AI Prompt Optimization Guide for 2024.” Accessed: Nov. 26, 2023. [Online]. Available: https://influencermarketinghub.com/ai-prompt-optimization/#toc-1

[37] “ChatGPT Prompt Engineering for Developers - DeepLearning.AI.” Accessed: Nov. 27, 2023. [Online]. Available: https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/

[38] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, “A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications,” Feb. 2024, Accessed: Jun. 06, 2025. [Online]. Available: https://arxiv.org/pdf/2402.07927

[39] “What is Prompt Engineering? - AI Prompt Engineering Explained - AWS.” Accessed: Jun. 06, 2025. [Online]. Available: https://aws.amazon.com/what-is/prompt-engineering/

[40] T. Kojima, S. Shane Gu, M. Reid Google Research, Y. Matsuo, and Y. Iwasawa, “Large Language Models are Zero-Shot Reasoners”.

[41] M. J. Posada-Ramírez, “Ontología y Lenguaje de la Realidad Social,” *Cinta de moebio*, vol. 50, no. 50, pp. 70–79, Sep. 2014, doi: 10.4067/S0717-554X2014000200003.

[42] “Web Ontology Language - Wikipedia.” Accessed: Jun. 06, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Web_Ontology_Language

[43] “OWL - Semantic Web Standards.” Accessed: Jun. 06, 2025. [Online]. Available: https://www.w3.org/OWL/

[44] M. U. Din *et al.*, “Ontology-driven Prompt Tuning for LLM-based Task and Motion Planning,” Dec. 2024, Accessed: Jun. 06, 2025. [Online]. Available: https://arxiv.org/pdf/2412.07493v1

[45] R. Nakada, W. Ji, T. Cai, J. Zou, and L. Zhang, “A Theoretical Framework for Prompt Engineering: Approximating Smooth Functions with Transformer Prompts,” Mar. 2025, Accessed: Jun. 06, 2025. [Online]. Available: https://arxiv.org/pdf/2503.20561

[46] “LLM Evaluation: Metrics, Methodologies, Best Practices | DataCamp.” Accessed: Jun. 06, 2025. [Online]. Available: https://www.datacamp.com/blog/llm-evaluation