



UNIVERSIDAD  
DE LA REPÚBLICA  
URUGUAY



FACULTAD DE  
INGENIERÍA

# Modelos de lenguaje para el análisis narrativo: personajes y relaciones en textos literarios

Informe de Proyecto de Grado presentado por

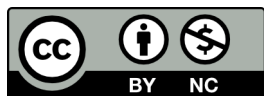
Francisco González Mayr y Juan Pablo Borja Cosentino

en cumplimiento parcial de los requerimientos para la graduación de la carrera  
de Ingeniería en Computación de Facultad de Ingeniería de la Universidad de  
la República

Supervisores

Dina Wonsever  
Diego Garat

Montevideo, 8 de junio de 2026



**Modelos de lenguaje para el análisis narrativo: personajes y relaciones en textos literarios** por Francisco González Mayr y Juan Pablo Borja Cosentino tiene licencia [CC Atribución - No Comercial 4.0](#).

# Agradecimientos

## Agradecimientos

Queremos expresar nuestro más sincero agradecimiento a nuestras familias y amigos por su apoyo incondicional a lo largo de todo este proyecto. Su paciencia ante nuestras ausencias, el ánimo en los momentos difíciles y la celebración de cada pequeño avance fueron fundamentales para sostener el esfuerzo cotidiano. Este trabajo también es de ustedes.

También queremos agradecer a nuestros tutores por su tiempo y acompañamiento a lo largo de todo el proceso.



# Resumen

El análisis computacional de textos literarios permite estudiar aspectos narrativos como el resumen, la clasificación, la detección de personajes y de vínculos entre ellos. Una forma particularmente efectiva de representar estas relaciones es a través de grafos, donde los personajes constituyen los nodos y las interacciones las aristas, formando redes sociales ficticias que revelan la estructura del relato.

Este proyecto propone la evaluación de técnicas de Procesamiento de Lenguaje Natural (PLN) y herramientas de análisis lingüístico para detectar personajes y sus vínculos en textos de ficción, con énfasis en el uso de modelos de lenguaje de gran tamaño sin entrenamiento adicional. Se abordan tres desafíos centrales: el reconocimiento de entidades nombradas, la resolución de correferencias, clave para agrupar diferentes menciones al mismo personaje, y la extracción de relaciones interpersonales con direccionalidad.

Para ello, se emplean herramientas conocidas en la comunidad de PLN y se desarrolla un sistema capaz de construir un grafo estático de relaciones a partir del texto. Se desarrolla un corpus experimental basado en historias de Sherlock Holmes, compuesto por personajes y alias, y sus relaciones, lo que permite evaluar la precisión del sistema propuesto. Este desarrollo se acompaña del estudio y comparación de distintas técnicas de ingeniería de *prompts*, así como de una revisión y evaluación de la literatura técnica vinculada a la temática. Además, se provee una visualización de los resultados que facilita la comprensión global del trabajo.

El resultado final es un prototipo funcional que permite identificar personajes y visualizar sus relaciones en textos literarios, generar resúmenes tanto del texto como de los personajes y realizar preguntas sobre el texto. Adicionalmente, el prototipo permite generar imágenes de los personajes reconocidos, y portadas para los textos analizados.

**Palabras clave:** PLN, redes de personajes, correferencia, modelos de lenguaje, grandes modelos de lenguaje (LLM) ingeniería de prompts.



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Alcance del Proyecto . . . . .	5
1.2. Estructura del Informe . . . . .	6
<b>2. Marco Teórico</b>	<b>7</b>
2.1. Redes de personajes y representación en grafos . . . . .	7
2.2. Modelos de Lenguaje . . . . .	10
2.2.1. Orientación de los Grandes Modelos de Lenguaje . . . . .	12
2.2.2. Manejo de grandes contextos . . . . .	13
2.3. Extracción de redes de personajes explícitas . . . . .	14
2.3.1. Predominancia de los modelos encoder-only . . . . .	15
2.3.2. Exploración de los modelos decoder-only . . . . .	15
2.3.3. Mitigación de problemáticas propias del dominio narrativo . . . . .	19
2.4. Corpus existentes . . . . .	19
2.5. Generación de Imágenes . . . . .	21
<b>3. Modelos de lenguaje para el análisis narrativo</b>	<b>23</b>
3.1. Descripción del problema . . . . .	23
3.2. Construcción del corpus propio . . . . .	24
3.3. Relaciones consideradas . . . . .	25
3.3.1. Relaciones personales . . . . .	25
3.3.2. Relaciones profesionales . . . . .	26
3.3.3. Relaciones de sospecha . . . . .	26
3.3.4. Relaciones de víctima . . . . .	27
3.3.5. Resumen . . . . .	28
3.4. Proceso de anotación . . . . .	28
3.4.1. Limitaciones y dificultades . . . . .	29
3.5. Exploración de enfoques y herramientas . . . . .	31
3.5.1. Experimentación con correferencia, spaCy-LLM, y alternativas . . . . .	33
3.5.2. <i>Embeddings</i> y recuperación . . . . .	33
3.5.3. Acercamiento a los grandes modelos de lenguaje . . . . .	34

<b>4. Implementación del prototipo</b>	<b>39</b>
4.1. Preprocesamiento y <i>embeddings</i>	40
4.2. Reconocimiento de entidades y correferencia	42
4.3. Generación de resúmenes	42
4.4. Generación de imágenes (FLUX DEV + ComfyUI)	45
4.5. Extracción de redes de personajes	46
4.5.1. Planteo y estrategia	46
4.5.2. Modelos	46
4.5.3. Variantes de <i>prompting</i>	47
4.5.4. Diseño de <i>prompts</i>	47
4.5.5. Consolidación final basada en LLM (deduplicación de personajes y relaciones)	48
4.5.6. Limitaciones	48
4.5.7. Resultado	50
4.5.8. Enfoques alternativos explorados tras la solución base	50
4.6. Interfaz del sistema: diseño, componentes y funcionalidades	51
4.6.1. Objetivos de diseño	51
4.6.2. Arquitectura tecnológica	51
4.6.3. Vistas y funcionalidades	52
4.6.4. Decisiones de implementación	54
4.6.5. Flujo de datos e integración con módulos	55
4.6.6. Limitaciones y trabajo futuro de la interfaz	56
<b>5. Experimentación</b>	<b>59</b>
5.1. Escenarios de evaluación	59
5.2. Partición del corpus y justificación	60
5.3. Metodología de evaluación	61
5.4. Resultados experimentales	62
5.5. Síntesis	67
<b>6. Conclusiones y trabajo futuro</b>	<b>69</b>
6.1. Conclusiones	69
6.2. Trabajo Futuro	71
<b>A. Prompts y corpus utilizados</b>	<b>89</b>
A.1. Prompts utilizados	89
A.1.1. Extracción de relaciones ( <i>Shots in System</i> )	89
A.1.2. Extracción de relaciones ( <i>Shots in Chat</i> )	92
A.1.3. Prompts del algoritmo SumAsk	94
A.1.4. Prompts del enfoque ChatIE	95
A.1.5. Resúmenes narrativos	96
A.1.6. Resúmenes por personaje (descripción física)	98
A.1.7. Resúmenes por personaje (identidad/rol)	100
A.1.8. Consulta contextual (chat sobre textos)	101
A.1.9. Reglas operativas para alias ( <i>alias policy</i> )	101
A.1.10. Prompts de correferencia con LLM	102

A.1.11. Consolidación final (fusión de alias y deduplicación) . .	106
A.2. Datos del Corpus . . . . .	107
A.2.1. Lista de textos del corpus . . . . .	107
A.2.2. Listado de partición del corpus . . . . .	109
A.2.3. Gráficas por cuento . . . . .	109



# Capítulo 1

## Introducción

El análisis computacional de textos narrativos, entendido como el conjunto de métodos y técnicas de procesamiento del lenguaje natural (PLN) que permiten modelar, interpretar y extraer información estructural de relatos escritos, ha cobrado relevancia en los últimos años por su capacidad de ofrecer perspectivas novedosas, tanto de forma individual como a nivel de colecciones completas de un autor, sagas o bibliotecas digitales [Jockers \(2013\)](#).

Entre las aplicaciones del análisis computacional destacan la generación automática de resúmenes, la identificación y caracterización de personajes y el estudio de redes sociales ficticias. En este marco, los grafos o redes de personajes, entendidos como estructuras en forma de grafo donde los nodos representan los personajes identificados en el texto narrativo y las aristas codifican relaciones predefinidas extraídas de dicho texto, se consolidan como una herramienta valiosa para representar y visualizar interacciones, permitiendo revelar la estructura y la dinámica de las relaciones que a menudo pasan inadvertidas en la lectura secuencial ([Labatut y Bost, 2019](#)).

Como se muestra en la [Figura 1.1](#), se presenta un ejemplo de grafo de personajes en el que cada vértice corresponde a un personaje y cada arista representa una de las cuatro relaciones predefinidas, diferenciadas mediante la etiqueta asociada a cada conexión.

De este modo, estos grafos permiten una comprensión más global del entramado de personajes y sus interacciones, sirviendo como base para los análisis desarrollados en este trabajo.

Dentro de las aplicaciones de estas redes, se encuentra la capacidad de deducir la importancia de los personajes de un texto narrativo, mediante la observación de las conexiones reconocidas. Esto permite deducir si un personaje es el protagonista, o si es un extra, al observar la cantidad y tipo de relaciones que posee, con el resto de los personajes ([Park y cols., 2012](#)). De igual forma, es posible identificar comunidades, realizando agrupamientos sobre la red de personajes, que permiten una vista profunda sobre la dinámica del texto narrativo ([Weng y cols., 2007](#)). En consonancia con la identificación de la relevancia de cada personaje, estas redes también permiten determinar el rol que desempeña cada uno

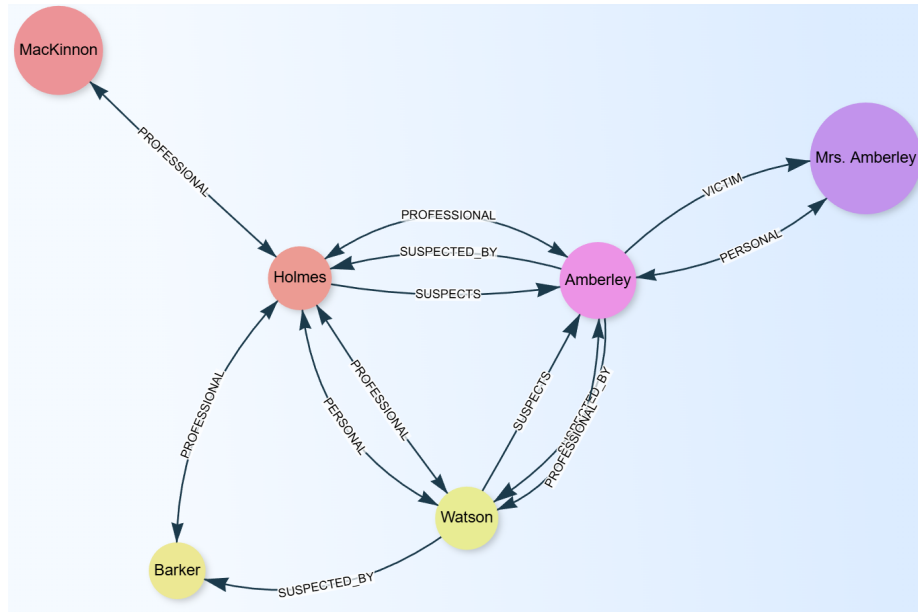


Figura 1.1: Grafo de Personajes, creado sobre el cuento *The Adventure of the Retired Colourman*, del libro *The Case-Book of Sherlock Holmes* escrito por Arthur Conan Doyle.

dentro de la narrativa. Dicho rol se entiende como un arquetipo previamente definido, entre los cuales se encuentran, por ejemplo, el rol de protagonista y el de antagonista.

Adicionalmente, las redes de personajes han sido utilizadas para evaluar el nivel de realismo de una obra de ficción, realizando una comparación de su topología con redes sociales reales, analizando el cumplimiento de las propiedades encontradas en estas últimas, como es el caso de [Stiller y cols. \(2003\)](#), que realiza un estudio sobre las obras de Shakespear y su capacidad de imitar redes sociales reales.

Además, la extracción automática de relaciones entre personajes en novelas no solo habilita análisis literario automatizado, sino que también impulsa la construcción de grafos de conocimiento y los sistemas de recomendación inteligentes ([Prado y cols., 2016](#)).

La creación de redes de personajes, está fuertemente relacionada con la extracción automática de relaciones, la cual a lo largo de los años ha sido objeto de numerosas investigaciones en distintos formatos de texto. Sin embargo, la efectividad de los métodos de extracción de relaciones dependen fuertemente de que los textos objetivos posean una estructura rigida, que facilite esta extracción, por lo que muchos estudios se han centrado en dominios semi estructurados, como lo son las noticias y enciclopedias ([Detroja y cols., 2022](#); [Ramos-Flores y Pinto, 2021](#); [Doddington y cols., 2004](#); [Walker y cols., 2006](#); [Riedel y cols., 2010](#);

Gurulingappa y cols., 2012)

En cuanto a la construcción de redes de personajes en narrativa, el enfoque predominante adoptado por los trabajos existentes se basa en la integración secuencial, en cascada de tareas de reconocimiento de entidades nombradas (NER por sus siglas en inglés), la resolución de correferencias y finalmente la extracción de relaciones (RE por sus siglas en inglés (Amalvy y cols., 2024a)).

Para la etapa de RE se utiliza la coocurrencia como metodología de extracción en donde se asume que existe una relación entre dos personajes cuando se encuentran mencionados dentro de una misma ventana de texto (Smeets, 2021). Este enfoque facilita la extracción, pero introduce ruido ya que la coocurrencia no garantiza interacción, y dificulta capturar relaciones de larga distancia, pues ampliar la ventana incrementa conexiones irrelevantes.

Por ejemplo, en la oración «Holmes permanecía en Londres, mientras Watson disfrutaba de unas vacaciones en el campo», la coocurrencia lleva a reconocer incorrectamente una relación, aunque no haya interacción entre ellos. En cambio, en «Holmes interrogó a Watson sobre los detalles del caso», la coocurrencia permite identificar correctamente una relación, dado que el texto describe una interacción explícita entre ambos personajes dentro de la ventana de coocurrencia.

Por otro lado, la identificación de relaciones explícitas como lo son las familiares, profesionales, o relaciones de acciones realizadas entre un personaje y otro, como por ejemplo, el personaje A golpeó al personaje B, conllevan diversas dificultades que exigen una comprensión semántica profunda, históricamente prohibitiva para métodos basados en reglas (Yan y cols., 2025; Bearman y Stovel, 2000; Ramos-Flores y Pinto, 2021; Canário y cols., 2025; Suen y cols., 2013).

Entre las principales dificultades se encuentran, en primer lugar, la necesidad de resolver adecuadamente fenómenos de referencia y correferencia, incluyendo alias, pronombres y descripciones indirectas, lo que vuelve muy complejo determinar con precisión qué personajes participan efectivamente en una relación. En segundo lugar, la ambigüedad semántica, la ironía, el discurso indirecto y la presencia de negaciones o modalizadores dificultan discernir si una acción describe realmente una relación entre personajes, si la niega o si la presenta como hipotética, algo especialmente difícil de capturar mediante patrones superficiales. Finalmente, muchas relaciones dependen de información distribuida a lo largo del texto, de inferencias contextuales o de conocimiento del mundo, lo que exige una comprensión semántica profunda y hace que su identificación resulte particularmente costosa para enfoques basados en reglas.

Por ejemplo, expresiones como «su viejo amigo regresó a Londres para ayudarlo en el caso» requieren inferir, a partir del contexto, quién es el poseedor y quién es el *viejo amigo*, lo que dificulta asociar de forma inequívoca la relación de amistad a un par concreto de personajes. De manera similar, en oraciones como «Holmes habría acusado a Watson si hubiera tenido pruebas suficientes», la presencia de condicionales y modalizadores impide interpretar la acusación como un hecho efectivo, aunque superficialmente aparezca acusación entre dos personajes. Asimismo, en pasajes donde se menciona que «el asesino del caso anterior observaba la escena desde la multitud», puede ser necesario conectar

esta descripción con información introducida muchos párrafos antes para identificar que se trata de un personaje previamente presentado, lo que ilustra la dependencia de relaciones explícitas respecto de información distribuida y de inferencias de largo alcance.

En este contexto, los avances recientes en modelos de lenguaje preentrenados a gran escala han mejorado sustancialmente la comprensión textual, gracias a que su entrenamiento masivo les permite modelar la semántica a partir de sutilezas textuales y aprovechar conocimiento general adquirido durante su entrenamiento (Chun y cols., 2025; Brown y cols., 2020).

Sin embargo, el dominio de textos narrativos, los cuales presentan desafíos propios del dominio, como lo son la presencia de alias, referencias indirectas y descripciones extensas, que dificultan la identificación y vinculación precisa de los personajes, presenta problemáticas particulares. Por ejemplo, un mismo personaje puede ser mencionado como Sherlock Holmes, el detective o el inquilino del 221B, lo que dificulta agrupar todas esas menciones bajo una misma entidad. De forma similar, referencias indirectas, como su viejo amigo o la muchacha del caso anterior, requieren inferir a qué personaje aluden a partir del contexto; asimismo, descripciones extensas que intercalan detalles del escenario o del pasado del personaje pueden separar en el texto las menciones clave, complicando la detección y vinculación precisa de las relaciones (Yan y cols., 2025; Labatut y Bost, 2019).

Por ello, se plantea la evaluación de técnicas de PLN y análisis de redes en conjunto con el uso de grandes modelos de lenguaje, o LLM por sus siglas en inglés, como componente especializado de comprensión, empleándolos para abordar precisamente el desafío semántico y superar las limitaciones inherentes a la coocurrencia.

Pese a las mejoras logradas por los grandes modelos de lenguaje en la comprensión de textos no estructurados, su adaptación para la resolución de extracción de relaciones, y particularmente la extracción de redes de personajes sin hacer uso de etapas de entrenamiento permanece comparativamente poco desarrollada. Esto refuerza la necesidad de su investigación en conjunto con técnicas establecidas de Procesamiento del Lenguaje Natural.

Con este propósito, el presente proyecto explora la construcción de redes de personajes en narrativa mediante la adaptación de grandes modelos de lenguaje, en conjunto con herramientas de PLN, y evalúa su efectividad sobre un corpus anotado durante el transcurso del proyecto, de cuentos de Sherlock Holmes.

Los libros de Sherlock Holmes, escritos por Arthur Conan Doyle a finales del siglo XIX y comienzos del XX, se componen de un conjunto de relatos o cuentos, cada uno de los cuales constituye un episodio autónomo dentro de la obra global. Esta estructura episódica implica que cada cuento presenta un caso relativamente independiente, de modo que puede leerse de forma aislada sin dificultar la comprensión de los siguientes, aunque todos mantengan un flujo narrativo general y compartan personajes recurrentes como el detective Holmes y el doctor Watson. Este corpus fue diseñado tanto para adaptar las técnicas de extracción como para medir su desempeño.

## 1.1. Alcance del Proyecto

El objetivo de este trabajo es la evaluación del uso de modelos de lenguaje para la extracción de redes de personajes en textos narrativos, donde para lograrlo se desarrolla un prototipo funcional capaz de identificar personajes y extraer relaciones de los textos, representándolas en forma de grafo en un flujo de trabajo unificado. Como objetivos específicos se plantea la implementación de las herramientas usuales de PLN, en conjunto con una experimentación sobre el uso de modelos de lenguaje para el análisis de textos con un enfoque en los LLM.

El sistema resultante busca facilitar la exploración estructural, y la extracción de información de obras literarias mediante visualizaciones que reflejen la dinámica social entre personajes, contribuyendo a la investigación de nuevas formas de análisis literario asistido por computadora. Adicionalmente se implementó un módulo capaz de generar imágenes tanto de portadas para los textos ingresados, como de los personajes identificados durante su procesamiento, para consolidar las capacidades de visualización del prototipo.

En particular el presente proyecto se centra en explorar la extracción de tres tipos de interacciones explícitas: menciones (un personaje habla sobre otro), acciones directas (un personaje actúa sobre otro) y afiliaciones (los personajes comparten una relación social explícita, como vínculos familiares o laborales).

En el marco del proyecto, el alcance se restringe al reconocimiento de personajes humanos y a la identificación de relaciones de tipo interpersonal. Para el análisis, se consideran únicamente aquellas relaciones inferibles del texto, sin importar si permanecen constantes o varían a lo largo de la narración, donde se considera que las relaciones son estáticas, de forma tal que si una relación fue reconocida en un parte del texto, es válida en el cuento completo.

Como caso de estudio, se construyen redes de personajes a partir de cuentos de Sherlock Holmes, en las que se modelan relaciones de tipo personal (por ejemplo, vínculos de amistad, parentesco o confianza, como la relación entre Holmes y Watson), profesional (como la relación entre un empleador y su empleado, o entre colegas involucrados en la investigación de un caso), de sospecha y sospechado (propia del contexto criminal trabajado, cuando un personaje identifica explícitamente a otro como sospechoso) y de víctima y victimario (también específica del dominio, cuando se indica que un personaje ha agredido o perjudicado a otro). Estas categorías abarcan, a un nivel más abstracto, los tipos de relaciones descritos en la literatura de redes de personajes que se consideran de mayor valor analítico, en particular aquellas asociadas a afiliaciones, acciones directas y menciones, es decir, relaciones fuertemente ligadas a la semántica del texto narrativo y no solo a su estructura sintáctica superficial.

La creación de redes de personajes se particiona utilizando el proceso secuencial propuesto por (Amalvy y cols., 2024a) con sus correspondientes subtareas de reconocimiento de entidades nombradas, resolución de correferencias y extracción de relaciones, donde se exploran las capacidades de adaptabilidad de los LLM mediante técnicas de prompting, e ingeniería de prompts. Entendemos por prompting la técnica de proporcionar instrucciones o ejemplos dentro de la

entrada de un LLM para guiar su comportamiento en una tarea específica, y por ingeniería de prompts el diseño sistemático de esas instrucciones (véase Sección 2.2.1). Se utilizaron herramientas consolidadas para el reconocimiento de entidades y la resolución de correferencias, incluyendo modelos de lenguaje de mediano porte, en conjunto con el uso de LLM. Finalmente para la extracción de relaciones, se aplicaron únicamente técnicas de prompting sobre LLM para una mayor exploración del área.

Entre las principales limitaciones técnicas se encuentran la necesidad de procesar textos extensos, que superan los límites contemplados por muchas herramientas; la complejidad adicional por utilizar las salidas de distintos módulos en cascada; y la escasez de corpus específicos centrados en redes sociales de personajes derivados de obras narrativas.

## 1.2. Estructura del Informe

En el capítulo 2 se presentan formalmente las redes de personajes, los tipos de relaciones reconocibles, los modelos de lenguaje y los grandes modelos de lenguaje, las técnicas de prompting y sus formas de aplicarlas a la creación de redes de personajes y extracción de relaciones. En el capítulo 3 se presenta la motivación para la creación del corpus anotado, su formato y metodología de anotación, el sistema propuesto y las diferentes variantes, para cada una de las etapas secuenciales implementadas para la creación de redes de personajes. En el capítulo 4 se describe el prototipo de sistema final producido y su arquitectura. En el 5 se discuten los experimentos realizados y los resultados obtenidos. En el 4.6 se presenta el sistema desde su interfaz gráfica, explicando sus funcionalidades y finalmente las vistas. Finalmente en el 6 se encuentran las conclusiones obtenidas y el trabajo a explorar a futuro.

## Capítulo 2

# Marco Teórico

Las redes de personajes ficticios y la teoría de las redes para el análisis de la trama de textos literarios, surgen impulsadas por la aceleración provocada en investigaciones basadas en textos literarios (Moretti, 2011), de forma de hacer uso de las técnicas que se encuentran dentro de la Extracción de Información (IE por sus siglas en inglés) en el ámbito del Procesamiento de Lenguaje Natural (PLN). Dentro de este marco y con respecto a la extracción de redes de personajes, la IE persigue la obtención de conocimiento estructurado, como lo es una red de personajes ficticios, desde textos no estructurados (Xu y cols., 2024).

Este capítulo aborda los diferentes conceptos teóricos utilizados a lo largo del proyecto, comenzando por el posicionamiento de la tarea a resolver dentro de la teoría en la sección 2.1, continuando por los enfoques y metodologías existentes para la creación de redes de personajes. Luego se presentan los conceptos teóricos más relevantes en la sección 2.2. Continuando por el estado del arte sobre las soluciones existentes en la creación de redes de personajes y el posicionamiento de los modelos de lenguaje en la sección 2.3. Acto seguido, en la misma sección se describe la aplicación de los grandes modelos de lenguaje y las problemáticas propias del dominio narrativo. Finalmente se concluye presentando los corpus existentes para el dominio del proyecto 2.4 , y de forma complementaria, una sección sobre la generación de imágenes 2.5.

### 2.1. Redes de personajes y representación en grafos

Formalmente describimos una red de personajes ficticios como un grafo  $G=(V,A)$  extraído de un texto narrativo (novela, cuento, guion, etc.), donde los nodos  $V$  representan personajes y las aristas  $A$  codifican interacciones o relaciones entre ellos.

Como se puede observar en la Figura 2.1, se presenta un ejemplo de grafo de personajes, realizado sobre el cuento *The greek interpreter* del libro *The Memoirs of Sherlock Holmes* de Arthur Conan Doyle.

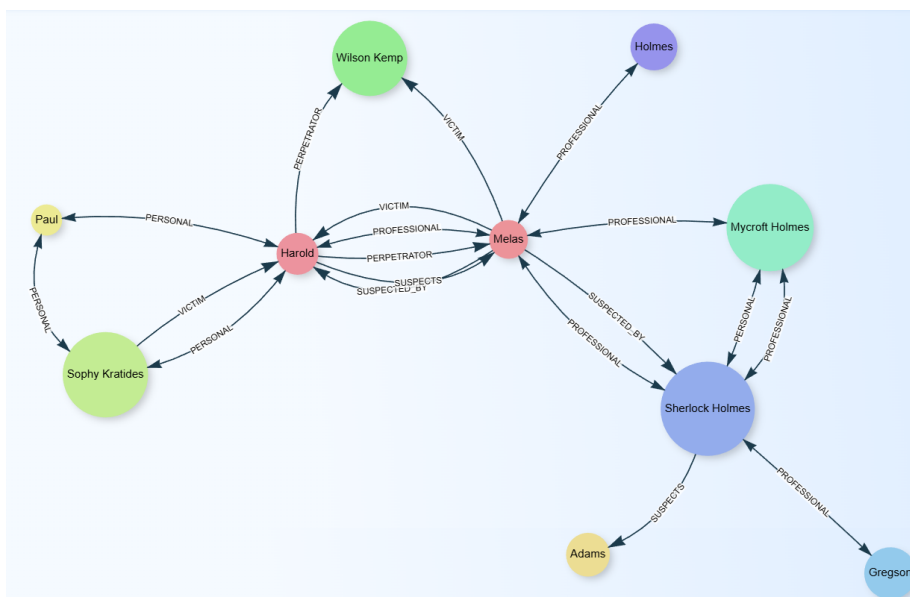


Figura 2.1: Vista de personaje: resúmenes específicos, relaciones incidentes y retrato.

La semántica de las aristas puede refinarse según su direccionalidad, según la interacción sea unilateral (aristas dirigidas) o recíproca (dos aristas opuestas o una arista no dirigida). Un caso claro de una relación recíproca es la de ser familiar de alguien, si A es familiar de B, el inverso también es cierto. Por otro lado como ejemplo de relación unilateral, se encuentra la relación *mención*, donde existe una relación de A a B si A menciona a B (Labatut y Bost, 2019).

La segunda forma de caracterizarlas es por el tipado de la relación: comención/coocurrencia, mención, acción directa, afiliación y conversacionales, entre otras. En el caso de comención o coocurrencia, el texto se segmenta en unidades narrativas oraciones, párrafos, capítulos y se infiere una relación entre A y B si los dos personajes A y B aparecen en la misma unidad. Esto puede aplicarse al coocurrir el nombre de dos personajes, ejemplificada por «“This is my friend Mr. Barker,“ said Holmes.», identificando una relación entre Mr. Barker y Holmes. Por otro lado, también podría aplicarse al coocurrir sus alias o correferencias, «Holmes took me aside before we left the room...», identificando una relación entre Holmes y el narrador, siendo Watson en este caso.

Una relación de mención de A a B existe cuando A menciona o refiere a B explícitamente en el texto. En la oración «“Cut out the poetry, Watson,“ said Holmes severely.», Holmes hace mención de Watson, y por lo tanto se puede reconocer una relación entre Holmes y Watson de tipo mención.

De esta forma, una relación de acción directa puede existir ya sea en conjunto mediante una acción donde ambos son partícipes, como *A y B pelean*, o acciones

donde un personaje A actúa sobre B, donde por ejemplo A asesina a B.

Adicionalmente, las relaciones de tipo *afiliación*, como las familiares o profesionales, son identificables como *A y B son parientes*. Este tipo de relaciones suelen aplicarse de forma recíproca *A es pariente de B* y por tanto *B es pariente de A*.

Finalmente las relaciones conversacionales se identifican cuando dos personajes entablan una conversación, donde se puede realizar la identificación mediante diálogos encapsulados entre comillas, o de forma menos estructurada ejemplificado mediante «Watson le dijo al detective que se fuera a casa».

Por otra parte, se puede asociar un peso o intensidad de la interacción a la relación reconocida, ya sea por frecuencia, duración o relevancia textual. Además, las relaciones se pueden caracterizar por su polaridad (vínculos positivos/negativos); y finalmente por su temporalidad, donde se introducen los grafos estáticos donde se agregan asociaciones a lo largo de toda la obra, y su contraparte, los dinámicos, que evolucionan a partir de esta temporalidad. Las redes de tipo estáticas presentan problemas debido a que las relaciones a identificar suelen cambiar y evolucionar conforme avanza la historia, problema que es agravado al analizar textos extensos y evitable mediante el uso de redes dinámicas. Estas redes introducen la necesidad de definir las ventanas temporales, que suelen ser inferidas mediante heurísticas y reglas, como es el caso de [DeBuse y Warnick \(2024\)](#), que utiliza la introducción y aparición de personajes para identificar cumulos de personajes que simbolizan eventos. Esto es agravado al observar que en los textos narrativos, foco de este proyecto, se permite narrar eventos pasados en capítulos posteriores, concluyendo el motivo por el cual pocos trabajos se han centrado en ellas.

Existen dos enfoques ampliamente utilizados para derivar las asociaciones entre personajes, el primero basado en relaciones de coocurrencias [Knuth \(1994\)](#); [Alberich y cols. \(2002\)](#); [Stiller y cols. \(2003\)](#); [Hettinger y cols. \(2015\)](#), y el segundo en relaciones explícitas, siendo estas de tipo menciones [Choi y Kim \(2007\)](#), acciones directas como las identificadas en [Bossaert y Meidert \(2013\)](#), afiliaciones [Bazzan \(2020\)](#) y relaciones conversacionales [Voloshinov y Gozhanskaya \(2009\)](#). La palabra explícitas hace referencia a que se busca identificar relaciones predefinidas, explicitadas previamente.

Muchas investigaciones han centrado su atención en este primer enfoque, donde su utilidad radica en la simplicidad y en su capacidad para ofrecer una vista global de la red. Sin embargo, conlleva imprecisiones conocidas ya que puede introducir falsos positivos; la coocurrencia entre dos personajes no implica que exista una interacción o relación entre ellos. Adicionalmente las redes resultantes presentan una falta de sentido semántico claro e introducen un posible sobredimensionamiento de personajes secundarios que coocurren con frecuencia.

El segundo enfoque modela las asociaciones como relaciones explícitas que buscan evidencias textuales de involucramiento entre personajes más allá de la coocurrencia. La extracción de redes de personajes con relaciones explícitas ha sido explorada en menor medida respecto de las investigaciones sobre coocurrencia ya que el reconocimiento de relaciones explícitas requiere de una comprensión semántica sobre el texto que complejiza su resolución. Es por ello que inicial-

mente investigaciones como [Agarwal y cols. \(2013\)](#) buscan construir sobre las coocurrencias, mediante la implementación de una etapa inicial de coocurrencia, donde se reconoce una instancia mediante pares de entidades reconocidos, seguido de la clasificación de la instancia utilizando la estructura del texto. Por otro lado, [Sudhahar y cols. \(2015\)](#) evitan hacer uso de la coocurrencia, mediante una extracción inicial de tripletas de tipo (sujeto, verbo, objeto), seguido de una ponderación de las entidades y las acciones asociadas y posteriormente la realización de un filtrado mediante reglas, utilizando la ponderación y la frecuencia de aparición de las entidades reconocidas. En [Trovati y Brady \(2014\)](#) se utilizan ambas metodologías donde inicialmente se utiliza una extracción de tripletas de tipo (sujeto, verbo, objeto) y para aumentar el recall de la estrategia, se consideran también las entidades que coocurren en el texto donde se evalúan utilizando reglas basadas en la gramática.

En [Labatut y Bost \(2019\)](#) se presenta como metodología recurrente en la creación de redes de personajes un proceso secuencial en cascada, donde se comienza por una etapa de reconocimiento de entidades nombradas (NER por sus siglas en inglés), seguido de una etapa de resolución de correferencias, continuando con una etapa de extracción de las relaciones (RE por sus siglas en inglés) predefinidas, y finalmente la creación de la red de personajes. Posteriormente formalizado en [Amalvy y cols. \(2024a\)](#), debido al frecuente uso de esta metodología por etapas.

Estos métodos basados en reglas y coocurrencias fallan en incluir información semántica del texto, al basarse puramente en reglas gramaticales y heurísticas. Con el objetivo de superar esta limitación, es que se motiva el uso de herramientas, y particularmente grandes modelos de lenguaje que permitan la inclusión de la información semántica presente en el texto, de la cuál hacer uso ha sido desafiante históricamente.

## 2.2. Modelos de Lenguaje

En esta sección se presentan los modelos de lenguaje, que son utilizados para modelar la semántica y sintaxis del lenguaje durante el proyecto, y particularmente los grandes modelos de lenguaje (LLM por sus siglas en inglés). Se presenta una definición de *prompting* y los principales tipos que existen, donde finalmente se alinean estas definiciones con la introducción del aprendizaje por contexto (ICL por sus siglas en inglés). Adicionalmente se presentan los *embeddings*, utilizados en el prototipo del sistema para realizar un manejo del texto en segmentos con información semántica asociada, para permitir la capacidad de preguntas y respuestas sobre el texto.

No existe como tal una definición globalmente aceptada sobre los modelos de lenguaje, y esto se debe a su heterogeneidad, variando tanto en su arquitectura, como en su entrenamiento. Una de las definiciones más aceptadas es, “Un modelo de lenguaje es un modelo de aprendizaje automático que predice las próximas palabras. Más formalmente, un modelo de lenguaje asigna una probabilidad a cada posible próxima palabra, o equivalentemente asigna una

distribución de probabilidad sobre todas las posibles próximas palabras. Los modelos de lenguaje también pueden asignar una probabilidad a una oración entera.” (Jurafsky y Martin, 2025). Los presenta como predictores del próximo token, lo cual tiene como problemática que no se adapta completamente al funcionamiento de los modelos de lenguaje como BERT (Devlin y cols., 2019), donde se utiliza un entrenamiento basado en el enmascaramiento y predicción del token enmascarado. Lo que caracteriza a los modelos de lenguaje es que aprenden una representación del lenguaje a partir de texto sin supervisión, con el objetivo de modelar la estructura y el significado del lenguaje natural.

Dado su alto desempeño, y la observación empírica de que este desempeño aumenta al aumentar la cantidad de parámetros, surgieron los grandes modelos de lenguaje (Kaplan y cols., 2020). Estos son directamente dependientes de la definición aceptada para Modelos de Lenguaje; Jeong y Ročková (2025) presenta una discusión sobre las diferentes características que vuelven estos modelos “grandes”, mencionando que los factores que se consideran en diversos artículos son los de el uso de la arquitectura Transformers, la presencia de miles de millones de parámetros y el entrenamiento utilizando una cantidad de datos masivos (W. X. Zhao y cols., 2023), mientras que adicionalmente Wiggins y Tejani (2022) contempla el entrenamiento de propósito general para su uso en diversas áreas, en contraste con los modelos tradicionales entrenados para aplicaciones específicas.

La arquitectura Transformer presentada inicialmente por Vaswani y cols. (2017) sustituyó la recurrencia por mecanismos de atención y habilitó el procesamiento paralelo de secuencias, lo que facilitó un salto en eficiencia y calidad frente a modelos recurrentes o convolucionales. Desde su introducción, se consolidó como base de tres familias hoy hegemónicas en PLN: modelos *encoder-only*, orientados a la comprensión y etiquetado de entradas; *encoder-decoder*, diseñados para mapear secuencias de entrada en salidas estructuradas; y *decoder-only*, especializados en generación autoregresiva de texto.

En términos funcionales, los *encoder-only* son preferibles para comprensión y etiquetado (clasificación de secuencias/token, NER, QA extractiva), donde las representaciones bidireccionales preentrenadas aportan señales robustas; teniendo un peor desempeño en generación abierta porque no están entrenados como modelos autorregresivos de salida libre.

Los *encoder-decoder* destacan cuando hay que mapear entradas en salidas estructuradas (traducción, resumen, paráfrasis, “texto-a-texto” en general); para tareas puramente discriminativas, suelen ser menos ventajosos que un *encoder-only* por simplicidad y eficiencia del enfoque representacional explorado en profundidad por Raffel y cols. (2020).

Los *decoder-only* brillan en generación autoregresiva y razonamiento guiado por prompting (*few/zero-shot, in-context learning*); en cambio, tienden a ser relativamente menos competitivos/eficientes en tareas de clasificación fina o comprensión semántica puntual frente a *encoders* entrenados para ello, donde esta segunda afirmación se ha debilitado debido al entrenamiento masivo realizado sobre ellos.

### 2.2.1. Orientación de los Grandes Modelos de Lenguaje

En orden de poder guiar a los grandes modelos de lenguaje en la resolución de diversos objetivos, es necesario conocer el concepto de *prompting* y sus principales características.

El *prompting* es una técnica que consiste en proporcionar instrucciones o ejemplos a un LLM dentro de su entrada (*prompt*) para guiar su comportamiento hacia una tarea específica. Esta metodología permite aprovechar el conocimiento pre entrenado del modelo sin necesidad de ajustar sus parámetros.

Las capacidades de adaptación de los LLM mediante esta orientación con un conjunto de instrucciones revelaron una capacidad propia de estos modelos: el aprendizaje en contexto o ICL por sus siglas en inglés. Se entiende como un paradigma en el que un modelo de lenguaje resuelve una tarea a partir de una instrucción y unos pocos ejemplos de demostración incluidos directamente en el *prompt*, sin actualizar sus parámetros. Esta formulación, hoy ampliamente utilizada, está sintetizada en revisiones recientes sobre ICL (Dong y cols., 2022). Existen variaciones que buscan optimizar la forma en que interactuamos con los LLM para obtener resultados más precisos y relevantes, donde se identifican:

- *Zero-Shot Prompting*: Se le da una instrucción al modelo para realizar una tarea sin proporcionarle ningún ejemplo de cómo debería hacerlo. El modelo se basa únicamente en su conocimiento obtenido en la etapa de entrenamiento para generar la respuesta.
- *Few Shot*: Se le da una instrucción al modelo junto con una lista de ejemplos de instancias correctas que permiten guiar al modelo en la forma en las que debe responder.
- *Chain of Thought*(CoT): Busca proveer a los modelos de lenguaje la capacidad de realizar una secuencia de pasos de “razonamiento” que permiten arribar a la solución final de un problema. Esto se hace típicamente mediante la ejemplificación de otras cadenas de pensamiento de forma *few shot* (J. Wei y cols., 2022).
- *Vanilla Prompting*: Si bien no existe una definición formal de *Vanilla Prompting* es un término al cual se hace referencia al no hacer uso de técnicas específicas, se lo llama *prompting* “ingenuo” y se lo toma como la base que utilizan los artículos al comparar otro tipo de técnicas o paradigmas de *prompting* (X. Wei y cols., 2023; G. Li y cols., 2023; S. Zhao y cols., 2020).

Dentro de las técnicas no especializadas de tipo *zero-shot*, para el alineamiento de los LLM en el seguimiento de instrucciones se distinguen las siguientes formulaciones para el estructuramiento de una *prompt*. Primeramente, otorgarle de forma explícita un rol específico al LLM que sirve de guía para las instrucciones que debe seguir, como por ejemplo, «Eres un experto en la extracción de relaciones...». Por otro lado, el estructuramiento de la salida, como tablas o JSON, ha obtenido empíricamente mejores resultados. Adicionalmente, descomposición

del problema en múltiples etapas, como fue mencionado previamente, resulta en un mejor alineamiento debido a que el LLM debe resolver problemas de menor complejidad (Zhou y cols., 2023; X. Wei y cols., 2023; G. Li y cols., 2023).

Continuando con la idea de descomponer el problema en múltiples etapas, existe una técnica basada en el refinamiento iterativo utilizando nuevamente un LLM. Inicialmente se instruye al LLM a resolver la tarea, y a continuación se toma esa solución inicial, y se instruye al LLM a refinarla, buscando errores o mejoras posibles, utilizando un formato de instrucción como «Dada la respuesta RESPUESTA, refinala siguiendo la LISTA\_DE\_INSTRUCCIONES» (Schulhoff y cols., 2025). Este tipo de técnica de ingeniería de prompts aumenta la consistencia en sus respuestas. De esta manera, otra de las técnicas se enfoca en instruirle al LLM explícitamente que otorgue una cita del texto de entrada, como justificación del resultado presentado, reforzando esta idea de consistencia basada en texto existente (Sarkar y cols., 2023).

La observación empírica que populariza este comportamiento de ICL se remonta a GPT-3, donde Brown y cols. (2020) mostró que escalar modelos autoregresivos potencia el desempeño *few-shot* mediante simples prompts con instrucciones y demostraciones, sin *fine-tuning*. En paralelo, Kojima y cols. (2022) constataron que los LLM en formato *zero-shot*, cuando el *prompt* induce pasos intermedios de razonamiento genéricos, como agregar la instrucción «pensemos el problema paso a paso», mejoran considerablemente su desempeño, lo que amplía el alcance del ICL más allá del uso de ejemplos, como es el caso de CoT con pasos de razonamiento adaptados a un objetivo específico. El rendimiento del ICL es altamente sensible al diseño del *prompt* (formulación, orden y selección de ejemplos, longitud y formato), y esta sensibilidad explica en parte la proliferación de técnicas de ingeniería de prompts para estabilizar resultados en tareas complejas, y el interés por la investigación de esta área aplicada a actividades de PLN (Dong y cols., 2022).

Otro de los factores a considerar en la construcción de *prompts* es el problema de *Lost in the Middle* donde (N. F. Liu y cols., 2023) presenta que los LLM tienen una degradación en el uso de la información en su ventana de contexto, la cual es la información que maneja un LLM en cualquier momento, donde se le adjudica mayor importancia a la información posicionada al comienzo y al final de la ventana de contexto, y la que se encuentra en el medio suele ser ignorada. Esto genera una problemática, donde el aumento de información e instrucciones en la *prompt*, no solo aumenta la complejidad del objetivo sino que también presenta el riesgo de que sea ignorada.

### 2.2.2. Manejo de grandes contextos

Como fue mencionado previamente, existe una limitación sobre la cantidad de información textual que puede manejar un LLM en un momento dado, limitación propia tanto por la arquitectura transformers presentada por Vaswani y cols. (2017), como por la degradación de información en su ventana contextual observado en N. F. Liu y cols. (2023). A través de la literatura esta problemática ha sido resuelta mediante el particionamiento del texto en segmentos, y una

adaptación en el uso de estos segmentos dependiendo de la tarea a resolver.

Como parte de esta solución, no se intenta proporcionar todo el texto al modelo en cada consulta, sino seleccionar y enviar únicamente los segmentos que se consideran más significativos para la pregunta planteada. Para poder identificar estos segmentos relevantes se requiere un mecanismo que permita comparar la consulta con cada fragmento de texto según su significado, y no solo según las palabras exactas que contienen.

Una de las soluciones más conocidas es el uso de *embeddings* para poder calcular distancias entre diferentes segmentos, donde se puede obtener una distancia semántica. Para poder acceder a los segmentos más relevantes al ingresar una pregunta, de forma de lograr una funcionalidad de preguntas y respuestas, se codifican representaciones vectoriales densas de los segmentos que mantienen información semántica (Schulhoff y cols., 2025). A estos vectores se los denomina *embeddings*, entendidos como representaciones vectoriales de unidades textuales (palabras o enunciados), donde la proximidad geométrica codifica relaciones semánticas y sintácticas. Surgen como alternativa a las codificaciones dispersas como *one-hot*, resolviendo su alta dimensionalidad y ausencia de estructura semántica (C. Zhang y cols., 2024). Dentro de los diferentes *embeddings* se encuentran las siguientes variaciones:

- Embeddings estáticos de palabra, como Word2Vec, GloVe y fastText, que asigna un único vector por tipo de palabra, capturando regularidades distribucionales.
- Embeddings de oración y de documento, produciendo vectores de longitud fija para unidades de tamaño variable.

Se busca que estos *embeddings* contengan información contextual y, por ende, su vector resultante sea dependiente del contexto, abordando así la polisemia y las dependencias de largo alcance.

### 2.3. Extracción de redes de personajes explícitas

Con el objetivo de realizar un análisis sobre textos literarios, en el correr de esta sección se evaluará el rol de los modelos de lenguaje para la resolución de las tareas de reconocimiento de entidades con nombre, resolución de correferencias y extracción de relaciones en textos o enunciados de diversos tipos de texto, y luego se entregará mayor énfasis sobre las técnicas utilizadas en el estado del arte que hacen uso de LLM para RE.

Las investigaciones sobre redes de personajes de relaciones explícitas no conversacionales, para la creación de un grafo estático se reducen drásticamente, como es evidenciado por Labatut y Bost (2019), revisión del estado del arte en la creación de redes de personajes.

### 2.3.1. Predominancia de los modelos encoder-only

Dado este panorama, procedemos a realizar un paneo general sobre el estado del arte de las tareas clásicamente involucradas en la creación de redes de personajes en el proceso secuencial mencionado previamente.

En general, los modelos de lenguaje han permitido mejoras significativas en la resolución de estas tareas de NER, resolución de correferencias y RE. Estos modelos, profundizados en la sección 2.2, se basan en la arquitectura Transformers, donde se encuentran tres variantes, los encoder-decoder, los cuales fueron los inicialmente introducidos por Vaswani y cols. (2017), los encoder-only, y los decoder-only. Si bien son muy parecidos, estos difieren en su desempeño, y presentan diferentes fortalezas y debilidades. En particular los modelos encoder-only como BERT, RoBERTa, SpanBERT (Devlin y cols., 2019; Y. Liu y cols., 2019; Joshi y cols., 2020) y derivados, han dominado históricamente los *benchmarks* de NER, correferencia y extracción de relaciones, sobresaliendo en tareas estructuradas, en formato supervisado.

En NER, las revisiones recientes, como Seow y cols. (2025), documentan el impacto sostenido de la arquitectura Transformer de tipo encoder-only luego de una etapa de *fine-tuning*, la cual se refiere a realizar un entrenamiento posterior sobre un modelo de lenguaje previamente entrenado con el fin de ajustarlo a una tarea en particular, como estrategia estándar para secuencias etiquetadas, con resultados competitivos y una amplia adopción en herramientas y marcos prácticos.

En cuanto a la resolución de correferencias, los avances de SpanBERT presentados por Joshi y cols. (2020) y sucesores marcaron el estado del arte en el conjunto de datos de evaluación de correferencias de CoNLL-2012, y trabajos posteriores, incluyendo Maverick, un proceso iterativo en cascada de resolución de correferencias presentado por Martinelli y cols. (2024), donde prevalece la dominancia de arquitecturas *encoder-only* eficientes, reforzando la idea de que la agregación contextual a nivel de documento se resuelve con mayor solidez utilizando estas arquitecturas.

De igual manera para la extracción de relaciones, Diaz-Garcia y Lopez (2025) presenta un análisis sobre el estado del arte en la resolución de RE, que contempló 137 artículos presentados en conferencias, concluyendo que existe una gran dominancia de métodos basados en BERT para alcanzar el estado del arte en RE, a la vez que reconoce el empuje de variantes que integran LLM.

### 2.3.2. Exploración de los modelos decoder-only

En paralelo, la investigación sobre *decoder-only* LLM (p. ej., GPT, LLaMA) para RE enfocado en documentos se ha intensificado. Por un lado, existen evaluaciones que señalan que, sin una etapa de ajuste de parámetros para la tarea, los LLM quedan por debajo de sistemas especializados en escenarios exigentes. Los LLM no son capaces de adaptarse a las diferencias provenientes de las posibles variantes en los datos (Swarup y cols., 2025), y por lo tanto el *fine-tuning* o adaptaciones ligeras siguen siendo necesarias. De forma de ejemplificar las li-

mitaciones de los modelos decoder, se contempla AutoRE presentado por [Xue y cols. \(2024\)](#). En el conjunto de datos de evaluación de RE por documentos, llamado DocRE, los LLM tienen resultados insatisfactorios. Se concluye, por lo tanto, que requieren estrategias adicionales para competir con modelos de lenguaje de tipo *encoder-only*.

Por otro lado, emergen enfoques híbridos prometedores, como es el caso de [X. Li y cols. \(2024\)](#) quienes proponen filtrar pares de entidades y delegar al LLM solo los candidatos plausibles, cerrando parte de la brecha frente a modelos tradicionales.

Estas líneas sugieren que el futuro de los modelos *decoder-only* es el de un componente final como refinadores o componentes generativos dentro de flujos que incorporan selección previa, calibración y, cuando corresponde, adaptación leve.

De igual forma se requiere de mayor investigación para poder asegurar la incapacidad de los LLM en la resolución total de RE. Las observaciones realizadas por [Wan y cols. \(2023\)](#) dan indicios sobre las capacidades inexploradas de los LLM sin etapas de *fine-tuning*. Este propone hacer uso puramente del aprendizaje en contexto, otorgándole ejemplos en un paradigma *few-shot* dinámicamente, donde se busca que los ejemplos entregados sean lo más cercanos posibles a la tarea a realizar y reformulando los enunciados de forma que se encuentren explícitamente las entidades y la tarea a realizar en el prompt, logrando resultados competitivos. Esta cercanía entre una entrada dada, y los ejemplos presentados, es realizada codificando los ejemplos con su correspondiente vector de *embeddings*, y el vector de *embeddings* correspondiente a la instancia evaluada. Finalmente se realiza un cálculo de distancias entre vectores para obtener los mejores ejemplos.

Adicionalmente se encuentran los enfoques basados en paradigmas de tipo *zero-shot*, donde únicamente se hace uso de reformulaciones de la tarea a resolver, explotando las capacidades de aprendizajes por contexto de los LLM mediante la construcción de prompts que guíen al modelo a tareas donde los LLM tengan un mejor desempeño.

Dentro de estas investigaciones se encuentra la técnica de Chain of Thought ([J. Wei y cols., 2022](#)), definida en la sección 2.2.1, la cual es una formulación donde se guía al modelo a generar una serie de pasos previo a realizar una decisión final.

Por otro lado, y con mayor enfoque en RE, se encuentra el enfoque QA4RE o Question Answering for Relation Extraction, presentado por [K. Zhang y cols. \(2023\)](#), donde se plantea reformular la tarea de RE como una de preguntas y respuestas. Se argumenta que las tareas de preguntas y respuestas suelen tener una mayor representación dentro de los conjuntos de datos de entrenamiento de los LLM, y se demuestra que esta alineación mejora los resultados en comparación con el vanilla prompting, o prompting ingenuo, donde no se hace uso de ninguna técnica más que el instruir al LLM a resolver la tarea.

Continuando con estas investigaciones, se encuentra ChatIE (Zero-Shot Information Extraction via Chatting with ChatGPT), presentado por [X. Wei y cols. \(2023\)](#), que hace uso de una reformulación de tipo divide y conquistarás,

partiendo de la observación de que los LLM tienden a presentar mejores resultados en tareas de extracción de información cuando se divide el problema en subtareas más simples y se aborda de manera interactiva. En el caso de la extracción de relaciones, la propuesta consiste en particionar la tarea en dos etapas: primero identificar los tipos de relaciones presentes en un enunciado y luego, a partir de la predicción, extraer las instancias concretas en un esquema de diálogo multi-turno de preguntas y respuestas.

Según lo argumentado por [X. Wei y cols. \(2023\)](#), el mantener todo el proceso en una misma ventana de contexto favorece el entendimiento del modelo y por ende su desempeño. Este enfoque se inspira en estudios previos como el previamente mencionado QA4RE, que reformulan la extracción de relaciones como un problema de Question Answering, y en trabajos que demuestran mejoras al introducir pasos intermedios de razonamiento como Chain of Thought prompting. En particular, se puede observar en la Figura 2.2, obtenida del artículo original, la adaptación de la tarea de extracción de relaciones utilizando este paradigma.

SumAsk (Revisiting Large Language Models as Zero-shot Relation Extractors) ([G. Li y cols., 2023](#)) aborda limitaciones detectadas en el vanilla prompting, el cual indirectamente intenta resolver de una sola vez tres problemas no triviales: detectar los pares de entidades relevantes, comprender el significado semántico de la relación y asignar la etiqueta correspondiente. Inspirado en ChatIE, QA4RE y en las mejoras observadas con Chain of Thought prompting, SumAsk propone dividir el proceso en tres etapas: (1) generar múltiples resúmenes (k-summaries) de las posibles relaciones entre un par de entidades en un enunciado; (2) reformular cada resumen en una pregunta de sí/no que mantenga la coherencia con el contexto; y (3) pedir al modelo que responda, consolidando las salidas mediante un voto por mayoría. Este mecanismo permite capturar relaciones múltiples superpuestas en un mismo contexto, aunque presupone que las entidades ya fueron identificadas de antemano, por lo que se enmarca más en el paradigma de clasificación de relaciones que en el de extracción de tripletas que se busca resolver en este proyecto. En la Figura 2.3 obtenida del artículo original, se puede ver un ejemplo sobre cómo funciona el algoritmo, y su procesamiento etapa por etapa.

Como se observa en la Figura 2.3, en la última etapa de SumAsk se encuentra un voto por mayoría para seleccionar si una relación es reconocida o no. Esto se debe a que los modelos de lenguaje, al ser modelos probabilísticos pueden producir salidas inesperadas, que pueden ser mitigadas al realizar múltiples ejecuciones. Entre las estrategias de robustecimiento, destaca Self-Consistency ([X. Wang y cols., 2022](#)), que muestrea múltiples cadenas de razonamiento y consolida sus respuestas, reduciendo la varianza inherente al muestreo de una única trayectoria, razonamiento que es trasladable a diversas tareas, como lo es la RE.

Debido a la capacidad de adaptación de los LLM junto con su constante mejora debido a la carrera tecnológica liberada por las grandes empresas, es de interés explorar su desempeño a la hora de generar redes de personajes ficticios. Siguiendo este hilo de pensamiento, es que ([Amalvy y cols., 2024b](#)) realizan una exploración sobre la implementación de LLM para evitar hacer uso del proceso

	Vanilla Prompt	Chat-based Prompt
1	<p><b>Question:</b> Suppose you are an entity-relationship triple extraction model. I'll give you list of head entity types: subject_types, list of tail entity types: object_types, list of relations: relations. Give you a sentence, please extract the subject and object in the sentence based on these three lists, and form a triplet in the form of (subject, relation, object).</p> <p>The given sentence is "Bono said that President Jacques Chirac of France had spoken eloquently of the need to support Africa , though he added that France had not yet come through with the resources ."</p> <p>relations: ['location-located_in', 'administrative_division-country', 'person-place_lived', 'person-company', 'person-nationality', 'company-founders', 'country-administrative_divisions', 'person-children', 'country-capital', 'deceased_person-place_of_death', 'neighborhood-neighborhood_of', 'person-place_of_birth']</p> <p>subject_types: ['organization', 'person', 'location', 'country']</p> <p>object_types: ['person', 'location', 'country', 'organization', 'city']</p> <p>In the given sentence, what triples might be contained? Please answer in the form (subject, relation, object):</p> <p>-----</p> <p><b>Expected Output:</b> [(Jacques Chirac, person-nationality, France)] <b>Output:</b> []</p>	<p><b>Question:</b> The given sentence is " Bono said that President Jacques Chirac of France had spoken eloquently of the need to support Africa , though he added that France had not yet come through with the resources ."</p> <p>List of given relations: ['location-located_in', 'administrative_division-country', 'person-place_lived', 'person-company', 'person-nationality', 'company-founders', 'country-administrative_divisions', 'person-children', 'country-capital', 'deceased_person-place_of_death', 'neighborhood-neighborhood_of', 'person-place_of_birth']</p> <p>What relations in the given list might be included in this given sentence? If not present, answer: none. Respond as a tuple, e.g. (relation 1, relation 2, .....):</p> <p>-----</p> <p><b>Expected Output:</b> (person-nationality) <b>Output:</b> (person-nationality)</p>
	<p>STAGE I</p>	
	<p>None</p>	<p><b>Question:</b> According to the given sentence, the two entities are of type ('person', 'country') and the relation between them is 'person-nationality', find the two entities and list them all by group if there are multiple groups. If not present, answer: none. Respond in the form of a table with two columns and a header of ('person', 'country'):</p> <p>-----</p> <p><b>Expected Output:</b> (Jacques Chirac, France) <b>Output:</b> (Jacques Chirac, France)</p>
	<p>STAGE II</p>	

Figura 2.2: Tabla donde se observa una comparación entre un enfoque directo, y el enfoque de ChatIE. Extraída de X. Wei y cols. (2023)

secuencial en cascada, con el objetivo de evitar la introducción de errores propios del procesamiento secuencial, donde cada etapa puede introducir errores que impactan en las etapas futuras. La investigación presenta dos acercamientos distintos, "LLM-Coref" que dado un segmento de texto, el LLM debe enmarcar las menciones de personajes junto con un identificador asociado, y por otro lado LLM-E2E el cual es un enfoque directo, donde se le pide al LLM que produzca en formato XML el grafo resultante. Finalmente, la faltante investigación sobre el uso de LLM para la generación de redes de personajes ficticios en formulaciones del problema con relaciones explícitas, justifica el enfoque de este proyecto sobre su aplicación en la etapa de RE, donde a diferencia de la investigación previa el proyecto hará uso del flujo clásico.

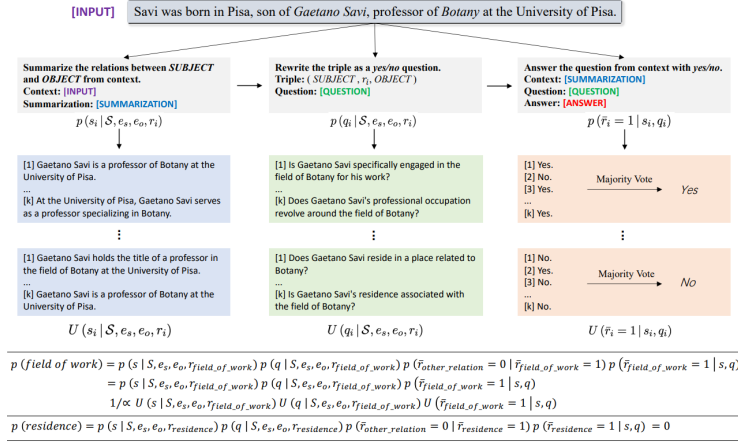


Figura 2.3: Ejemplificación de la formulación *SumAsk*. Extraída de G. Li y cols. (2023)

### 2.3.3. Mitigación de problemáticas propias del dominio narrativo

Debido al dominio de textos utilizados en el proyecto, se presenta el problema del manejo de contexto, dado que la capacidad de los grandes modelos de lenguaje tiene un límite predefinido. Surge, entonces, la necesidad de particionar la información proporcionada al modelo, y por tanto el procesamiento global de los textos narrativos. Esto, en conjunto con el problema presentado en *Lost in the Middle* N. F. Liu y cols. (2023), explicado en la sección 2.2.2, donde se observa que la información e instrucciones otorgadas a un modelo se degradan, resulta en una necesidad de segmentar los textos, y procesarlos de manera iterativa. Esto inevitablemente introduce una pérdida de información sobre relaciones de larga distancia existentes en el texto.

Finalmente, luego de la construcción de una red de personajes, es posible que se introduzcan personajes identificados como dos entidades diferentes, pero que se refieren a una misma persona, debido al uso de alias y posibles errores en la coreferencia, agravado por el dominio narrativo, donde el uso de alias suele ser frecuente. Para solucionarlo, (Chun y cols., 2025) propone una etapa de postprocesamiento sobre la red de personajes inicialmente construida, donde un LLM recibe una descripción de los personajes en conjunto con un resumen del texto, y las relaciones reconocidas, y este se encarga de realizar una unión sobre los grupos de personajes que detecta como una misma entidad.

## 2.4. Corpus existentes

Con el objetivo de evaluar la construcción de redes de personajes a partir de relaciones explícitas de mayor valor analítico, se decide excluir las relacio-

nes conversacionales por su menor contenido semántico y su carácter predominantemente sintáctico respecto de las anteriores. En una fase inicial se efectuó un estudio exploratorio de corpus anotados de narrativa orientados a redes de personajes; este análisis evidenció la inexistencia de un corpus específicamente diseñado para extraer redes a partir de múltiples tipos de relaciones explícitas que habilite la evaluación de la construcción de un grafo estático.

Dado estos resultados insatisfactorios, se procede a explorar los corpus de extracción de relaciones genéricos, que ha sido abordada en diferentes contextos de investigación, y ha dado lugar a múltiples corpus anotados. Estos conjuntos de datos suelen clasificarse de acuerdo con tres criterios: la longitud de las instancias, la cantidad de relaciones anotadas y el tipo de tarea de extracción (véase Tabla 2.1).

En primer lugar, los corpus por longitud de instancia pueden estar compuestos por enunciados o por documentos. En los primeros, cada instancia corresponde a una sola oración; en los segundos, a uno o varios párrafos. Incluso en estos últimos casos, sin embargo, la extensión es limitada y no se aproxima a la complejidad de los textos narrativos extensos que se busca analizar en este proyecto.

En segundo lugar, se distinguen corpus con una única relación por instancia y aquellos con múltiples relaciones. Los primeros resultan útiles para tareas de clasificación, mientras que los segundos se acercan más al escenario de extracción de tripletas en textos complejos.

Por último, los corpus existentes también se dividen por tipo de tarea: algunos están orientados a la clasificación de relaciones predefinidas entre entidades ya delimitadas, mientras que otros plantean la extracción completa de tripletas (entidad1, relación, entidad2) directamente a partir del texto. Aunque los primeros pueden adaptarse a tareas de extracción, el hecho de proveer las entidades subraya una diferencia sustancial, ya que el modelo se ve *guiado* en lugar de realizar la detección de manera autónoma.

Entre los corpus considerados se encuentran ACE05 (Walker y cols., 2006), CoNLL04 (Roth y Yih, 2004), NYT (Riedel y cols., 2010), SemEval2010 (Hendrickx y cols., 2010), TACRED (Y. Zhang y cols., 2017), TACREv (Alt y cols., 2020) y Re-TACRED (Stoica y cols., 2021), todos basados en instancias de enunciado (véase Tabla 2.1). En la categoría de documentos, se destacan ADE (Gurulingappa y cols., 2012), SciERC (Luan y cols., 2018), DocRED (Yao y cols., 2019), Re-DocRED (Tan y cols., 2022) y DocGNRE (J. Li y cols., 2023), los cuales sí contemplan múltiples relaciones y, en algunos casos, incorporan coreferencias (véase Tabla 2.1). Sin embargo, pese a estas características, presentan limitaciones para nuestro propósito: la mayoría está conformada por textos científicos, biomédicos o periodísticos, con esquemas de relación específicos de esos dominios. Además, incluso en los corpus de documentos, la longitud de los textos sigue siendo reducida en comparación con narrativas literarias extensas.

En consecuencia, si bien estos recursos representan hitos importantes en la investigación de extracción de relaciones, ninguno resulta adecuado para evaluar el desempeño del sistema en el dominio narrativo literario que constituye el foco de este trabajo.

**Tabla 2.1:** Conjuntos de datos para extracción de relaciones, agrupados por largo de instancia, número de relaciones por instancia y tipo de tarea.

Conjunto	Largo de instancia	Nº de relaciones	Tipo de RE
ADE			
SciERC			
DocRED	DOCUMENTO	Múltiples	Extracción de trípletas
Re-DocRED			
DocGNRE			
SemEval2010			
TACRED	ENUNCIADO	Una	Clasificación de relaciones
TACREV			
Re-TACRED			
ACE05	ENUNCIADO	Múltiples	Clasificación de relaciones
NYT			
CoNLL04	ENUNCIADO	Múltiples	Extracción de trípletas

## 2.5. Generación de Imágenes

Como complemento visual, este trabajo incorpora la generación de imágenes condicionada por texto para elaborar una portada del libro y retratos de personajes. En términos generales, estos modelos aprenden una correspondencia texto–imagen que permite sintetizar imágenes coherentes con descripciones en lenguaje natural y se utilizan aquí solo con fines ilustrativos y de comunicación de resultados (N. Zhang y Tang, 2024).

Dentro de este campo, pueden distinguirse dos enfoques principales: aquellos basados en la similitud textual–visual, que utilizan representaciones conjuntas del texto y la imagen, como CLIP (Radford y cols., 2021), y los que integran un modelo de lenguaje para una comprensión semántica más profunda de la descripción textual (L. Wang y cols., 2025). Ambos métodos buscan mejorar la coherencia entre el texto de entrada y la imagen generada, aunque difieren en el grado de comprensión lingüística que alcanzan. La selección del modelo específico empleado en este trabajo se detalla en la sección de resultados.



## Capítulo 3

# Modelos de lenguaje para el análisis narrativo

En este capítulo se presenta el sistema desarrollado para la identificación de personajes y la extracción de relaciones en textos literarios extensos, así como el *corpus* propio construido para su evaluación. El proceso de diseño fue iterativo: se realizaron exploraciones sucesivas, se obtuvieron resultados parciales y se ajustaron componentes hasta converger en el flujo de la aplicación final.

Además de describir dicho flujo, se introduce la creación del corpus anotado orientado a redes de personajes —con foco en relaciones explícitas—, detallando los textos utilizados, los tipos de relaciones reconocidas y la justificación de su inclusión, junto con el procedimiento de anotación y el resultado obtenido. Finalmente, se discuten las dificultades y limitaciones encontradas durante la anotación.

### 3.1. Descripción del problema

El objetivo del proyecto no es la creación de una herramienta por sí misma, sino la validación y fundamentación de un método que permita: (i) identificar personajes, (ii) extraer vínculos entre ellos y (iii) representarlos como una red interpretable a lo largo de la obra.

Trabajar con modelos de lenguaje en narrativa extensa presenta tres desafíos principales:

1. **Ventanas de contexto limitadas.** No resulta viable transmitir la obra completa a un gran modelo de lenguaje (LLM); adicionalmente, el fenómeno *lost in the middle* reduce la atención a pasajes intermedios (N. F. Liu y cols., 2023).
2. **Extracción con modelos *decoder-only*.** La literatura ofrece menos evidencia sistemática sobre su robustez en tareas donde las relaciones no apa-

recen formuladas de forma directa, sino que deben reconstruirse a partir de indicios implícitos, inferencias narrativas o dependencias contextuales.

- 3. Complejidad narrativa y evolución temporal de vínculos.** En literatura, las relaciones pueden cambiar con el desarrollo de la trama. Por ejemplo: una amistad (PERSONAL) puede tensarse y transformarse en una relación de tipo SOSPECHA/SOSPECHADO\_POR; un vínculo PROFESSIONAL “superior–subordinado” puede virar si interviene un VICTIMARIO y uno de ellos se convierte en una VICTIMA; o un trato distante puede revelar parentesco y reetiquetarse como PERSONAL. Capturar estas transiciones requiere una representación dinámica de la red.

A partir de estos elementos, se adoptó un flujo que: (a) segmenta la obra en fragmentos, (b) reconoce las entidades que aparecen en cada segmento y (c) realiza inferencia iterativa sobre los fragmentos, consolidando resultados en una red de personajes. En las siguientes secciones se documenta el recorrido que condujo a este diseño, indicando qué se probó, qué se descartó y con qué criterios.

## 3.2. Construcción del corpus propio

Ante la falta de un corpus apropiado, se decidió crear un conjunto de datos propio, diseñado específicamente para abordar la tarea de extracción de relaciones entre personajes en textos literarios. La construcción del corpus respondió a tres criterios principales: (i) la necesidad de contar con textos de suficiente longitud, enfocados en narrativa; (ii) la conveniencia de trabajar con materiales de dominio público que permitieran su uso sin restricciones legales (iii) la capacidad de contemplar las relaciones que se consideran de valor.

Tras analizar distintas opciones, se optó por utilizar los 56 cuentos de Sherlock Holmes de Arthur Conan Doyle. Estos textos presentan varias ventajas: mantienen un estilo narrativo relativamente homogéneo, contienen un número recurrente de personajes principales y secundarios, y giran en torno a un conjunto estable de tipos de relaciones —personales, profesionales y criminales— que resultan representativos y recurrentes. La lista detallada de cuentos seleccionados se incluye en el apéndice A.2 de este informe.

Adicionalmente, en la etapa inicial del proyecto se empleó *The Sound of Thunder* de Ray Bradbury como texto piloto para las pruebas: calibración de segmentación, NER con SpaCy, gráficos exploratorios y primeros ensayos de extracción. Una vez consolidado el flujo, la evaluación sistemática se centró en el corpus de Sherlock Holmes.<sup>1</sup>

---

<sup>1</sup>Corpus de relaciones interpersonales basado en 56 cuentos de Sherlock Holmes de Arthur Conan Doyle, con anotaciones manuales de seis tipos de relaciones explícitas. Disponible en: <https://gitlab.fing.edu.uy/proyecto-final2/corpus-sherlock-holmes>

### 3.3. Relaciones consideradas

Con el objetivo de obtener un esquema de anotación manejable, consistente y ajustado al dominio de los relatos de Sherlock Holmes, se definió un conjunto reducido de tipos de relación. Estas categorías buscan capturar vínculos explícitos expresados en el texto, menciones directas, acciones y afiliaciones, sin fragmentar en etiquetas demasiado finas que dificulten la anotación fiable o produzcan clases con muy pocos ejemplos.

En este trabajo se consideraron cuatro tipos principales: PERSONAL, PROFESIONAL, SOSPECHA y VÍCTIMA. A continuación se describe el alcance de cada una, con ejemplos tomados del corpus y criterios prácticos utilizados durante la anotación. Las categorías no son excluyentes: un mismo par de personajes puede presentar más de un tipo de relación.

#### 3.3.1. Relaciones personales

La categoría PERSONAL agrupa vínculos de cercanía entre personajes: relaciones familiares (padres, madres, hijos, hijas, hermanos, hermanas, matrimonios, compromisos, parientes cercanos), amistades, convivencia prolongada o lazos afectivos y de confianza claramente marcados en el texto. Se modela como relación bidireccional: si el relato establece un vínculo personal relevante entre  $A$  y  $B$ , se representa como una relación compartida en el grafo.

Un caso emblemático es el de Sherlock Holmes y el Dr. Watson. Más allá de colaborar en investigaciones, numerosos pasajes los presentan como amigos cercanos. Por ejemplo, en *The Red-Headed League*, Watson introduce a Holmes como:

“I had called upon my friend, Mr. Sherlock Holmes, one day in the autumn of last year...”

La formulación “my friend” es una evidencia directa de un vínculo personal. De modo análogo, vínculos familiares se anotan cuando el texto los declara explícitamente, como en *The Greek Interpreter*, donde Holmes presenta a su hermano:

“This is my brother Mycroft.”

Las relaciones de pareja o matrimonio explícito también se incluyen en PERSONAL. En *A Scandal in Bohemia*, por ejemplo, la narración describe la boda entre Irene Adler y Godfrey Norton; el relato deja claro que pasan de una relación previa a un vínculo conyugal formal, por lo que el par se anota como relación PERSONAL.

En lugar de diferenciar etiquetas específicas como AMISTAD, MATRIMONIO o HERMANOS, se optó por agrupar estos casos bajo una única categoría PERSONAL. Esto simplifica la anotación, reduce ambigüedades y permite concentrarse en la presencia o ausencia de vínculos estrechos relevantes para la estructura de la red.

### 3.3.2. Relaciones profesionales

La categoría PROFESIONAL recoge relaciones laborales, jerárquicas o de colaboración sostenida: empleador–empleado, servicio doméstico, encargos profesionales explícitos, asociaciones de trabajo u otros vínculos funcionales claramente identificables en el texto. También se representa como relación bidireccional en el grafo, aunque los papeles narrativos sean asimétricos (jefe, subordinado, cliente, etc.).

Por ejemplo, en *The Red-Headed League*, Jabez Wilson describe a Vincent Spaulding como su asistente en la casa de empeños:

“I have a small pawnbroker’s business at Saxe-Coburg Square . . . I only keep one [assistant] . . . His name is Vincent Spaulding . . . I should not wish a smarter assistant.”

El texto establece de forma explícita una relación de trabajo estable entre ambos, que se anota como PROFESIONAL. En *The Musgrave Ritual*, el vínculo amo–mayordomo entre Reginald Musgrave y Brunton se indica de manera directa:

“Of these servants the one who had been longest in our service was Brunton the butler. . .”

Asimismo, se incluyen encargos específicos cuando el contrato profesional es claro, como en *The Engineer’s Thumb*, donde el coronel Lysander Stark contrata al ingeniero Victor Hatherley para un trabajo técnico concreto.

Un caso particular es la dupla Sherlock Holmes–Dr. Watson. Además del vínculo PERSONAL ya mencionado, muchos relatos muestran una relación PROFESIONAL: Watson acompaña a Holmes como asistente, médico y cronista oficial de los casos. En nuestro esquema, esto se refleja permitiendo que el mismo par de personajes tenga simultáneamente relaciones PERSONAL y PROFESIONAL, capturando así su doble rol en la narrativa.

### 3.3.3. Relaciones de sospecha

Las relaciones de SOSPECHA son dirigidas y representan el hecho de que un personaje considera a otro como posible autor o partícipe de un crimen. Se anota un par ordenado  $(A, B)$  cuando el texto indica explícitamente que  $A$  sospecha, acusa o señala a  $B$ .

En *The Blue Carbuncle*, por ejemplo, Holmes confronta a James Ryder como responsable del robo de la joya:

“The game’s up, Ryder . . . I have almost every link in my hands, and all the proofs which I could possibly need. . .”

Aquí se anota una relación de SOSPECHA (y, en este punto, prácticamente acusación) desde Holmes hacia Ryder. En *Silver Blaze*, la policía sostiene la hipótesis de que Fitzroy Simpson es culpable:

“The police imagine, I take it, that this Fitzroy Simpson, having drugged the lad . . . opened the stable door and took out the horse. . .”

Este fragmento justifica una relación de SOSPECHA desde la policía hacia Simpson. Esta categoría captura la dinámica investigativa del género policial: quién sospecha de quién y en qué dirección fluye esa sospecha, lo que añade estructura orientada al grafo de relaciones.

### 3.3.4. Relaciones de víctima

La categoría VÍCTIMA representa la relación causal entre la víctima de un crimen y su autor o autores. También es dirigida: se anotan pares con roles diferenciados (víctima, perpetrador) cuando el texto ofrece evidencia suficiente de que un personaje es responsable del daño sufrido por otro. No se asume simetría ni se completa automáticamente la relación inversa.

En este tipo de relaciones es especialmente frecuente que la información relevante esté dispersa a lo largo del relato. A menudo el crimen o la muerte se introduce en una escena inicial y sólo más adelante, mediante confesiones, deducciones de Holmes o reconstrucciones finales, se identifica al culpable. Durante la anotación fue necesario localizar estos distintos momentos y confirmar que el texto vincula de forma clara víctima y perpetrador.

En *The Boscombe Valley Mystery*, por ejemplo, la narración establece que Charles McCarthy ha sido asesinado:

“He had been murdered.”

Más adelante, John Turner confiesa haberlo matado:

“I did it, Mr. Holmes . . . I struck him down.”

La combinación de ambos fragmentos permite anotar una relación VÍCTIMA entre McCarthy (víctima) y Turner (perpetrador).

En *The Adventure of the Speckled Band*, la muerte de Julia Stoner se atribuye finalmente al plan del Dr. Grimesby Roylott mediante una serpiente venenosa; en *The Sign of Four*, la muerte de Bartholomew Sholto se asocia a Tonga (bajo la dirección de Jonathan Small) a partir de la escena del crimen y la confesión posterior (casos similares se observan en *The Adventure of Black Peter* y *The Abbey Grange*, donde la resolución del caso termina identificando explícitamente al responsable).

Estos ejemplos muestran que, para anotar una relación VÍCTIMA, no basta con la mera presencia de un crimen en el texto: se exige que la autoría esté claramente establecida por la narración (confesión, prueba concluyente o explicación de Holmes). Este criterio fue clave para asegurar consistencia en la anotación manual y anticipa las dificultades que enfrenta la extracción automática de relaciones causales en narrativas extensas y no lineales.

### 3.3.5. Resumen

En conjunto, las cuatro categorías se definieron de acuerdo con las necesidades del dominio: PERSONAL y PROFESIONAL condensan afiliaciones estables y frecuentes; SOSPECHA captura las hipótesis e imputaciones propias de la investigación policial; y VÍCTIMA incorpora la dimensión delictiva y causal. Este diseño busca equilibrar detalle semántico y viabilidad práctica, ofreciendo un esquema de relaciones suficientemente expresivo para el análisis de redes de personajes sin introducir una explosión de etiquetas difíciles de anotar o aprender.

## 3.4. Proceso de anotación

La construcción del corpus requirió una lectura detallada de cada cuento y la identificación de las entidades y sus alias (por ejemplo, *Holmes* y *Mr. Holmes*). Se registraron las relaciones entre personajes en forma de tripletas del tipo *Personaje1, Personaje2, Relación*.

El proceso se llevó a cabo de manera manual, sin herramientas de etiquetado asistido: se leyeron todos los relatos y, ante la primera aparición de un personaje con nombre completo, se fijó su nombre canónico (por ejemplo, *Sherlock Holmes*). Las menciones posteriores se fueron incorporando como alias cuando correspondía (por ejemplo, *Mr. Holmes* o simplemente *Holmes*), manteniendo la trazabilidad dentro de cada cuento. Para decidir qué debía considerarse alias se establecieron reglas operativas que contemplan, entre otros aspectos, el tratamiento de títulos y honoríficos, la preferencia por la forma completa cuando está disponible, la normalización tipográfica (apóstrofos y guiones) y la homogeneización de iniciales y variantes ortográficas según la edición. Estas reglas no sólo guiaron la anotación, sino que además se reutilizan más adelante en el sistema para la unificación de alias, a fin de evitar nodos duplicados en el grafo de relaciones (véanse 4.2 y 4.5.5). La especificación completa de dichas reglas se incluye en el apéndice (véase A.1.9).

Los datos se almacenaron en archivos CSV, lo que facilitó su integración posterior en los módulos de la aplicación. Como resultado, se obtuvieron dos archivos por cada uno de los cuentos de Sherlock Holmes: uno correspondiente a las relaciones extraídas y otro al listado de personajes junto con sus respectivos alias, con el objetivo de permitir la correferenciación de las entidades reconocidas. En las Figuras 3.1 y 3.2 se muestran fragmentos representativos de cada uno de estos archivos, ilustrando respectivamente la estructura de las relaciones anotadas y la información utilizada para la normalización de menciones. Cabe destacar que las anotaciones se realizaron de manera independiente para cada cuento.

El corpus, es decir, los CSV de relaciones y de alias por cuento, junto con una breve documentación de criterios y algunos datos estadísticos— se encuentra disponible en el repositorio institucional.<sup>2</sup>

---

<sup>2</sup><https://gitlab.fng.edu.uy/proyecto-final2/corpus-sherlock-holmes>

```

Personaje1,Personaje2,Relacion
Sherlock Holmes,Josiah Amberley,PROFESSIONAL
Josiah Amberley,Sherlock Holmes,PROFESSIONAL
Mrs. Amberley,Josiah Amberley,VICTIM
Josiah Amberley,Mrs. Amberley,PERPETRATOR
Dr. Ray Ernest,Josiah Amberley,VICTIM
Josiah Amberley,Dr. Ray Ernest,PERPETRATOR

```

Figura 3.1: Fragmento del archivo CSV de relaciones correspondiente a uno de los cuentos de *Sherlock Holmes*, donde se observan las entidades involucradas y el tipo de relación anotada.

```

Nombre,Aliases
Sherlock Holmes,[my friend:Mr. Sherlock Holmes:Mr. Holmes:Holmes:S. H.]
Josiah Amberley,[the prisoner:our client:the old fellow:old Amberley:our unfortunate client:Mr. Josiah Amberley:Mr. Amberley:Amberley:The old colourman]
Watson,[my astute friend:my long-suffering friend:Dr. Watson:Narrator:my dear Watson:dear fellow]

```

Figura 3.2: Fragmento del archivo CSV de personajes y *aliases*, utilizado para la correferenciación de entidades reconocidas en cada cuento.

En la Figura 3.3 se aprecia, a nivel agregado, que las relaciones de afiliación (PERSONAL + PROFESSIONAL) concentran cerca de tres cuartas partes de los vínculos del corpus. En promedio, PROFESIONAL aporta alrededor de 40–45 % del total y PERSONAL cerca de 30–35 %, mientras que las categorías criminales (VÍCTIMA/PERPETRADOR y SOSPECHAS) suman aproximadamente 20–25 %.

Además de los porcentajes, los conteos absolutos por cuento muestran magnitudes consistentes con ese reparto: típicamente se observan 5–10 relaciones PERSONAL por cuento, 4–10 PROFESIONAL, entre 1–3 VÍCTIMA/PERPETRADOR y 0–3 SOSPECHAS. Estas cifras ofrecen un panorama general de densidad relacional por relato y confirman que las tramas están dominadas por vínculos de afiliación, con una fracción menor dedicada a dinámicas criminales.

Los desgloses detallados (por libro y por cuento) con los conteos de cada tipo de relación se incluyen en el apéndice; allí pueden consultarse las gráficas y valores específicos para cada relato (A.2.3).

### 3.4.1. Limitaciones y dificultades

El proceso de construcción enfrentó varios desafíos. Por un lado, los textos están escritos en un inglés propio de finales del siglo XIX, con palabras y usos del lenguaje que pueden no estar contemplados en el entrenamiento de los LLM modernos. Esto incluye léxico monetario y de oficios, como “*fifty guineas*” en *The Engineer’s Thumb*; vocabulario de transporte, como “*hansom*” en *The Man*

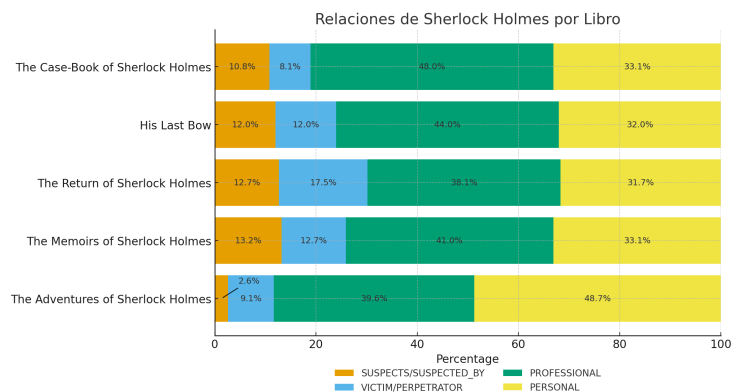


Figura 3.3: Relaciones anotadas por tipo por libro.

with the Twisted Lip; fórmulas de cortesía arcaizantes, como “Pray, take a seat”; y giros cotidianos hoy poco frecuentes, como “You are looking for lodgings” en *A Study in Scarlet*.

Por otra parte, la presencia de tratamientos y títulos —por ejemplo, “Mr. Jabez Wilson” o “Dr. Roylott”— junto con variantes ortográficas como *Holmes’s* y *Holmes’* favorece la duplicación de entidades por alias y complica la normalización.

Por otro lado, la narrativa literaria introduce una gran cantidad de relaciones implícitas, cuya correcta identificación requiere considerar el contexto completo de la obra. Por ejemplo, en *The Adventure of the Speckled Band*, la relación de víctima entre Helen Stoner y su padrastro Grimesby Roylott solo se entiende al llegar al desenlace, ya que gran parte del cuento está dedicada a la construcción de sospechas y pistas indirectas. De forma similar, en *A Scandal in Bohemia*, la conexión personal y de confianza entre Holmes y Watson no se declara explícitamente en cada episodio, pero subyace a lo largo de la narración y es esencial para interpretar sus interacciones. Otro caso ocurre en *The Adventure of the Bruce-Partington Plans*, donde la relación profesional de Mycroft Holmes con el gobierno británico solo queda clara tras varias menciones distribuidas en el texto, lo que obliga al lector (y al sistema) a integrar la información dispersa para reconstruir la relación.

Además, algunos vínculos se desarrollan a lo largo de varios cuentos, como la amistad y colaboración constante entre Holmes y Watson, que constituye un elemento estructural de toda la saga. Para evitar sesgos de dependencia intertextual, se decidió recortar el alcance del corpus a las relaciones presentes en cada relato individual, sin importar la continuidad que pudieran tener en otros textos.

Finalmente, es importante señalar que, dado que los cuentos de Sherlock Holmes son de dominio público y ampliamente conocidos, existe una posible superposición con los datos de entrenamiento de los modelos utilizados. Este sesgo potencial debe ser tenido en cuenta al interpretar los resultados experimentales.

### 3.5. Exploración de enfoques y herramientas

Previo a fijar definir el *corpus* definitivo, se realizaron pruebas exploratorias para comprender el comportamiento de los componentes básicos del sistema y estimar su utilidad para etapas posteriores. En esta fase preliminar se ensayó el reconocimiento de entidades con *spaCy* y se diseñaron visualizaciones que capturan la dinámica temporal de aparición de personajes a lo largo del texto. A partir de estos aprendizajes se ajustaron criterios y parámetros que luego se consolidaron en el *corpus* anotado. A continuación se describe esta etapa exploratoria.

Se realizaron experimentos iniciales con SpaCy (`en_core_web_trf`<sup>3</sup>) para reconocimiento de entidades nombradas. Como punto de partida se tomó la propuesta del *Scatter Plot of Entities* de DeBuse y Warnick (2023), que introduce una visualización orientada a describir el flujo narrativo a partir de la aparición de personajes.

En nuestra adaptación, cada punto del gráfico representa una mención de una entidad de tipo PERSON. El eje *x* codifica la posición de esa mención en el texto (según el orden narrativo, medida en índices de tokens), mientras que el eje *y* corresponde al orden de primera aparición de la entidad: el primer personaje que aparece en el relato ocupa la fila superior, el segundo la siguiente, y así sucesivamente. Todas las menciones posteriores del mismo personaje se representan en la misma fila, de modo que la distribución horizontal de puntos refleja en qué tramos del texto vuelve a intervenir.

Esta construcción permite identificar, de un vistazo, personajes introducidos temprano con presencia sostenida (filas densas a lo largo del eje *x*), personajes episódicos concentrados en una única zona del relato y apariciones tardías asociadas a subtramas locales. En este trabajo la visualización se utilizó como herramienta exploratoria para comprender la dinámica de aparición de personajes y para motivar decisiones posteriores.

Asimismo, se evaluó el flujo con el modelo en español de SpaCy (`es_core_news_lg`) para reconocimiento de entidades, pero los resultados resultaron ser muy insuficientes.

En la práctica aparecieron detecciones erróneas y duplicaciones sistemáticas, incluso aplicando ciertos filtros, por lo que para solucionar esto se restringieron las entidades a las etiquetadas como PERSON cuya categoría gramatical fuese PROPN, y se exploraron heurísticas simples de normalización, como preferir formas canónicas sin tratamiento: por ejemplo, elegir “Eckels” en lugar de “Mr. Eckels”.<sup>4</sup>

Si bien estas medidas redujeron parte de los errores, persistieron problemas

---

<sup>3</sup>`en_core_web_trf`. Este es un modelo de SpaCy basado en *transformers* para inglés general, que incluye segmentación, etiquetado morfosintáctico, dependencias y reconocimiento de entidades nombradas, y se utilizó aquí para la detección de entidades de tipo PERSON.

<sup>4</sup>De forma tentativa se aplicó la siguiente regla: si una mención contiene un tratamiento u honorífico (Mr., Mrs., Dr., Sir, Lady, Inspector, etc.) y existe una subcadena más informativa que coincide con el nombre o apellido principal, se adopta esta última como forma canónica, tras normalizar mayúsculas y signos tipográficos. La estrategia se descartó al observar pérdidas y casos ambiguos.

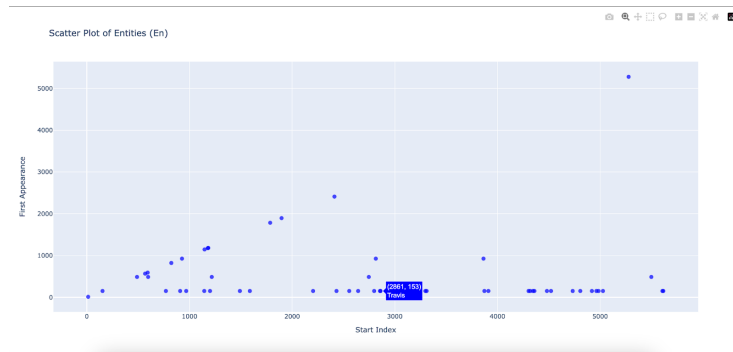


Figura 3.4: *Scatter Plot of Entities* adaptado para *A Sound of Thunder*. Cada punto representa una mención de una entidad de tipo PERSON. El eje  $x$  indica la posición de la mención en el texto y el eje  $y$  el orden de primera aparición de la entidad.

característicos del dominio narrativo. En textos como *A Sound of Thunder*, las formas con y sin tratamiento se separaban como si fueran personajes distintos (“Mr. Travis” frente a “Travis”, “Mr. Eckels” frente a “Eckels”), aparecían organizaciones o marcas mal clasificadas como personas (por ejemplo, “Time Safari, Inc.” o “Safari” en contextos con verbos de habla), y términos como “Tyrannosaurus Rex” se fragmentaban, promoviendo “Rex” a entidad de tipo persona. También se detectaron errores ligados a diálogos: en líneas con comillas, guiones o puntuación compleja, el modelo tomaba palabras finales o intermedias como nombres propios, o ignoraba parcialmente menciones afectadas por apóstrofes tipográficos.

En el corpus de los cuentos de Sherlock Holmes, estos fenómenos se amplificaron. Variantes como “Mr. Holmes”, “Holmes” y “Sherlock Holmes” fueron tratadas como entidades diferentes; lo mismo ocurrió con “Dr. Watson” y “Watson”, o con nombres acompañados de rangos y títulos, donde elementos como “Inspector”, “Mrs.”, “Lady” o “Sir” aparecían aislados como personas independientes. Se observaron además topónimos o apellidos compuestos segmentados de forma incorrecta (como “Baker Street” etiquetado como persona en construcciones posesivas, o apellidos como “St. Clair” y “Van Saar” partidos y reutilizados como entidades espurias), y dificultades claras con alias narrativos: casos como “Hosmer Angel” y “James Windibank” o las variantes de “Moriarty” e “Irene Adler” quedaban representados como nodos separados en ausencia de una consolidación posterior. A esto se sumó la tendencia a promover descripciones genéricas a nombres propios (“the Client”, “the Woman”, “the Doctor”) en posiciones salientes del enunciado, así como errores recurrentes en diálogos con guiones largos o saltos de línea, que fragmentaban o duplicaban menciones.

En conjunto, estos patrones motivaron dos decisiones: por un lado, tratar con cautela el uso directo de modelos de NER generales en narrativa literaria; por otro, justificar la incorporación de reglas de alias y módulos de consolidación

específicos en las etapas posteriores del flujo de extracción de relaciones.

Durante esta etapa se identificaron duplicaciones de personajes debidas a alias y variantes superficiales (p.ej., *Irene Adler / Miss Adler, Mr. Holmes / Sherlock Holmes*). Este fenómeno afecta la interpretación del *scatter plot* al fragmentar las menciones de una misma entidad en múltiples puntos. Más adelante se evaluó e implementó un procedimiento específico de detección y unificación de alias, con el fin de consolidar estas variantes en un único nodo canónico (véase 4.2).

### 3.5.1. Experimentación con correferencia, spaCy-LLM, y alternativas

Para la extracción de entidades también se probó *spaCy-LLM* (Explosion AI, 2025) con *few-shot prompting*. Se observó variabilidad alta ante modificaciones mínimas en los ejemplos y, además, la restricción de contexto resultó poco práctica para narrativas extensas. Por ello, se retornó al flujo estándar de SpaCy.

En cuanto a correferencia, se evaluaron *AllenNLP* (Allen Institute for Artificial Intelligence (AI2), 2025, 2022) y *NeuralCoref* (Hugging Face / spaCy Universe, 2025), encontrándose conflictos de dependencias y problemas de estabilidad. Se consideró el modelo *Coref-mT5-XXL* de Google (estado del arte), pero su tamaño (~13B parámetros) lo tornó inviable para la infraestructura disponible. En consecuencia, se adoptó una correferencia ligera (normalización y reemplazos acotados), priorizando estabilidad y trazabilidad. Para esto se utilizó *spaCy-experimental* (Explosion AI, 2023), un módulo que reúne funcionalidades en fase de prueba, el cual finalmente pudo integrarse dentro de un entorno aislado, aunque con limitaciones importantes en términos de estabilidad y precisión.

### 3.5.2. *Embeddings* y recuperación

Dado que los LLM no pueden procesar indefinidamente texto en una sola pasada, y que existe un límite práctico de ventana de contexto que varía entre modelos y versiones, en lugar de enviar el cuento completo al modelo optamos por un esquema de recuperación. En este esquema, el texto se divide en fragmentos, se indexa con *embeddings* y, ante cada consulta de extracción, se seleccionan sólo los  $k$  fragmentos más relevantes para construir el *prompt*. El flujo completo es: fragmentación → vectorización → búsqueda por similitud → construcción del *prompt*.

Para dar una idea de escala en unidades comprensibles para el lector, los relatos del corpus tienen, en general, longitudes del orden de las decenas de miles de caracteres. Muchos cuentos se ubican típicamente en un rango aproximado de 30 000 a 80 000 caracteres. Aún cuando un texto pudiera caber dentro de la ventana de contexto de cierto modelo, incluirlo íntegro suele ser contraproducente: (i) incrementa el costo y el tiempo de inferencia; (ii) diluye la evidencia relevante entre pasajes poco informativos; y (iii) puede degradar la calidad por efectos de

atención no uniforme en contextos largos, fenómeno mencionado anteriormente denominado como *lost in the middle* (N. F. Liu y cols., 2023).

El uso de recuperación permite (i) mantener el *prompt* dentro de límites de contexto razonables sin depender de la ventana específica de cada modelo, (ii) focalizar la evidencia pertinente a la pregunta (reduciendo el ruido) y (iii) modular el tamaño del contexto a través de la cantidad de fragmentos y el tamaño de cada segmento.

En nuestro flujo, comenzamos utilizando *Universal Sentence Encoder* (USE) para la indexación semántica y posteriormente migramos a `text-embedding-3-small`, lo que simplificó la compatibilidad con los modelos de generación empleados y ofreció una mejor alineación semántica en nuestra colección.

Para asegurar cortes por tokens, se empleó *Tiktoken* (OpenAI, 2025b). Cada segmento se representó en  $R^n$  y, durante la inferencia, se comparó el *embedding* de la consulta con los segmentos por distancia coseno, recuperando únicamente evidencia relevante (*RAG*). Este esquema redujo costos, disminuyó ruido y brindó mayor control sobre el contexto presentado al modelo. Se adoptó la similitud coseno como criterio de comparación entre *embeddings* por las siguientes razones: (i) mide el ángulo entre vectores y, por tanto, es invariante a la escala (robusta a diferencias de norma debidas a longitud del segmento o a efectos de tokenización); (ii) al centrarse en la dirección del vector, favorece la proximidad semántica frente a variaciones de magnitud y (iii) combinada con normalización  $\ell_2$  previa<sup>5</sup>, reduce sesgos que el producto punto introduce a favor de vectores de mayor norma.

En la práctica, ante cada extracción: (1) se formula la consulta (con el *prompt* y las instrucciones de salida), (2) se recuperan los  $k$  fragmentos más relevantes según similitud en el espacio de *embeddings*, y (3) se compone el mensaje al LLM sólo con esos fragmentos y metadatos mínimos (p. ej., origen, posición). Esta estrategia resultó más estable que inyectar el cuento completo, incluso cuando el tamaño del texto no excedía la ventana de ciertos modelos.

### 3.5.3. Acercamiento a los grandes modelos de lenguaje

Se ensayaron variantes cuantizadas de *Llama 2* (7B/13B) (Touvron y cols., 2023) en entorno local para posibilitar su ejecución con hardware disponible<sup>6</sup>. No obstante, la cuantización —imprescindible en nuestro caso para poder ejecutarlo— introdujo degradaciones de calidad y el consumo de memoria/tiempo resultó elevado.

En este contexto, y dado que el entrenamiento (o ajuste fino) de modelos especializados de gran porte demanda recursos de cómputo y presupuesto prohibitivos para el alcance de este proyecto, se priorizó un enfoque sin entrenamiento

---

<sup>5</sup>En la práctica, se normalizan los *embeddings* de segmentos y consultas a norma unitaria antes de la indexación/búsqueda.

<sup>6</sup>La cuantización es un proceso de compresión que reduce la precisión numérica de los pesos de un modelo (por ejemplo, de 16/32 bits a 8, 4 o incluso 2 bits) con el fin de disminuir el uso de memoria y acelerar la inferencia, aunque a costa de posibles pérdidas de calidad.

adicional. Esta decisión se alinea con la evidencia reciente de que los LLMs exhiben capacidades de resolución de tareas mediante *in-context learning* (ICL), es decir, a partir de instrucciones y ejemplos en el *prompt*, sin modificar pesos del modelo (Brown y cols., 2020). Además, varias de las tareas aquí evaluadas constituyen escenarios aún poco explorados para el ajuste fino tradicional, lo que reduce el retorno esperado de entrenar un modelo dedicado frente a alternativas de *prompting* y orquestación.

Por estas razones se migró al uso de modelos de OpenAI en la nube —primero `gpt-3.5-turbo` y luego `gpt-4o-mini` (OpenAI, 2025a)—, ejecutados íntegramente en la infraestructura de OpenAI. Este cambio eliminó las restricciones de cómputo locales (sin necesidad de cuantizar, con ventanas de contexto y *throughput* gestionados por el proveedor) y mejoró la estabilidad de inferencia, la latencia y la reproducibilidad, manteniendo un mejor equilibrio costo-precisión para nuestro *pipeline*. En síntesis, se optó por explotar *ICL* y buenas prácticas de diseño de *prompts*/recuperación en lugar de entrenar modelos específicos.

### Módulo de consultas (Prueba de concepto)

Se implementó una prueba de concepto para consultas libres sobre las obras: tras la carga y el preprocesamiento, el sistema ejecuta recuperación por *embeddings* y construye un *prompt* con los fragmentos más relevantes para que el LLM responda (*RAG*: segmentación → recuperación → inferencia). Este módulo sirvió para validar el flujo de punta a punta y calibrar parámetros de recuperación (*top-k*, tamaño de segmento), pero no fue el foco principal del estudio, por lo que se mantuvo como prueba de concepto.

A nivel de diseño, se utilizó un *system prompt* que fija el rol del modelo y una plantilla de usuario que inyecta la pregunta y los segmentos recuperados. Las plantillas completas pueden consultarse en el apéndice (A.1.8, List. A.24 y A.25); a continuación se muestra un fragmento ilustrativo:

Listing 3.1: Fragmento del *system prompt* para consulta contextual

```
You are an expert at extracting information from long works using only
retrieved segments.
Answer succinctly and ground your answer in the provided excerpts.
```

Listing 3.2: Fragmento de la plantilla de usuario (primer turno)

```
Question: <PREGUNTA_DEL_USUARIO>
Context:
<SEGMENTO_1>
<SEGMENTO_2>
...
```

En ejecución, el módulo reemplaza `<PREGUNTA_DEL_USUARIO>` por la consulta del lector y `<SEGMENTO_i>` por los pasajes recuperados, evitando enviar el texto completo y reduciendo ruido en la respuesta.

## Evolución del *prompting* para extracción de relaciones

La construcción de las *prompts* siguió un proceso incremental. En primera instancia se diseñaron *prompts de estilo libre*: instrucciones breves y poco restrictivas, sin rol explícito ni formato de salida predefinido (p. ej., sin plantilla JSON o campos obligatorios), que pedían al modelo “extraer relaciones” a partir de un segmento y devolverlas en texto natural. Este enfoque favoreció la rapidez de iteración, pero mostró dos efectos indeseados: (i) variabilidad en la forma de la respuesta (listas en prosa, viñetas, órdenes cambiantes de las tripletas) y (ii) mayor propensión a salidas incompletas o “alucinaciones” puntuales, al no imponer restricciones sobre el contenido ni su estructura.

Posteriormente, y a la luz de resultados preliminares, se adaptaron y reforzaron las plantillas con insumos tomados de la literatura (p. ej., guías de *few-shot/shots in system* vs. *shots in chat*, especificación de esquemas de salida y restricciones negativas), incorporando:

1. Estructuración fuerte del *system prompt*: se formalizó el rol del modelo, el *input format* y el *output format* con JSON válido y campos obligatorios (Véase 2.2.1 y A.1).

Listing 3.3: System prompt (núcleo) con JSON y evidencia

```
You extract PERSON-PERSON relations from retrieved segments.
Return VALID JSON: {"relationships":[{"entity1, entity2,
relationship_type, supporting_text}, ...]}.
If there is no explicit evidence in the provided text, return {"
relationships": []}.
"supporting_text" MUST quote or paraphrase the exact fragment
that justifies the relation.
```

(Plantilla completa en el Apéndice, A.1.1.)

2. Definiciones operativas de las etiquetas (PERSONAL, PROFESSIONAL, SUSPECTS, SUSPECTED\_BY, VICTIM, PERPETRATOR), incluyendo la especificación de relaciones bidireccionales cuando corresponde y las *exclusiones* relevantes, es decir, los casos que deben descartarse explícitamente (por ejemplo, evitar relaciones autorreferenciales o incompatibles con el tipo definido), con el fin de alinear criterios (A.1.1, A.1.2).
3. Versiones con *Shots in System* y *Shots in Chat*: se incluyen ejemplos mínimos para guiar y estabilizar el comportamiento del modelo. Los *Shots in System* son ejemplos integrados directamente en el mensaje del sistema (antes de cualquier interacción), mientras que los *Shots in Chat* son ejemplos proporcionados dentro del propio diálogo entre usuario y asistente. Ambos sirven como demostraciones explícitas del formato y del nivel de detalle esperado en la extracción (A.1.2, A.4).

Listing 3.4: Few-shot (Shots in Chat) mínimo y estabilizador

```
# User (snippet):
Segments: "Anne and George are best friends."

# Assistant (expected JSON):
{"relationships":[{"entity1":"Anne","entity2":"George",
"relationship_type":"PERSONAL",
"supporting_text":"Anne and George are best friends."}]}
```

(Versión extendida en [A.1.2.](#))

4. Condicionamiento con lista de entidades: se proporciona al modelo una lista previa de entidades candidatas (normalizadas y verificadas) para restringir la extracción a pares plausibles. Este mecanismo reduce falsos positivos, evita que el modelo invente nombres inexistentes y fuerza la coherencia con el conjunto de personajes ya identificados en etapas anteriores.

Listing 3.5: Condicionamiento con lista de entidades (entrada del usuario)

```
1. Relationships: [PERSONAL, PROFESSIONAL, SUSPECTS,
SUSPECTED_BY, VICTIM, PERPETRATOR]
2. Entity candidates (PERSON): ["Sherlock Holmes","John Watson",
"Irene Adler","Mycroft Holmes", ...]
3. Segments (top-k retrieved): <SEGMENTO_1> <SEGMENTO_2> ...
```

5. Reglas explícitas contra alucinación y cobertura: si no hay evidencia, devuelve vacío; *supporting\_text* obligatorio; y separación clara entre evidencia explícita y razonamiento contextual (Véase [2.2.1](#)).

Listing 3.6: Reglas anti-alucinación y cobertura

```
RULES:
- Do NOT guess relations. Only include items grounded in the
provided segments.
- For each relation, cite "supporting_text" from the segment;
otherwise exclude it.
- If nothing is supported, return {"relationships": []}.
```

En paralelo, se experimentó con esquemas propuestos en trabajos específicos para extracción: *SumAsk* ([A.1.3](#); Listings [A.6–A.9](#)) y *ChatIE* ([A.1.4](#); Listings [A.10–A.12](#)), integrando *k-summaries*, preguntas de sí o no por relación o ciclos detección → extracción (Véase [2.3.2](#)). Estas variantes, aunque conceptualmente atractivas, no superaron a las plantillas guiadas con definiciones y ejemplos en nuestro dominio narrativo, y quedaron como líneas comparativas.

El resultado de esta evolución es un conjunto de *prompts* más guiados y reproducibles, con formato de salida estable, *system prompts* claros y *user prompts*

parametrizables (listados en el apéndice [A.1](#)). Esta transición, de *prompts* propios a plantillas adaptadas siguiendo la bibliografía, permitió mejorar la consistencia, reducir la variabilidad de las respuestas y reforzar el anclaje en evidencia textual mediante el campo `supporting-text`.

### **Sobre resúmenes e imágenes**

Durante la fase exploratoria se verificó la viabilidad de generar resúmenes iterativos por segmentos y de utilizar descripciones por personaje como insumo para modelos *text-to-image*. Por tratarse de componentes estables del artefacto final, su diseño y justificación se documentan en la Arquitectura del sistema (véanse [4.3](#) y [4.4](#)).

## Capítulo 4

# Implementación del prototipo

El prototipo implementa de extremo a extremo el flujo para identificar personajes y extraer relaciones explícitas en narrativa extensa, priorizando tres principios: viabilidad con componentes simples y fácilmente sustituibles, trazabilidad y control de costos, gracias a un contexto acotado mediante recuperación y al uso selectivo del LLM. El sistema sigue un flujo desacoplado que incluye la ingesta y normalización de textos, la segmentación por tokens, la indexación semántica mediante *embeddings*, la recuperación de fragmentos relevantes, el reconocimiento de entidades con manejo de alias y correferencia opcional, la extracción iterativa de relaciones con consolidación y deduplicación, y finalmente, la generación de salidas, como resúmenes, grafos y recursos auxiliares.

La implementación expone interfaces claras entre etapas mediante formatos estructurados y contratos explícitos de entrada y salida. Además, incluye una parametrización centralizada de los principales hiperparámetros del sistema, como el tamaño de los segmentos, el valor de  $k$  en la recuperación, las variantes de *prompting* y el número de pasadas. Sobre esta base, la arquitectura final logra un equilibrio entre simplicidad y extensibilidad, de modo que cada módulo puede reemplazarse o ampliarse, por ejemplo, con nuevos modelos de *embeddings* o con plantillas alternativas de extracción— sin afectar al resto del sistema.

La arquitectura se organiza en una cadena de procesamiento con responsabilidades nítidas y acoplamiento laxo entre módulos (Fig. 4.1). Partiendo de la ingesta de documentos, el sistema ejecuta: (i) preprocesamiento y segmentación por tokens; (ii) representación semántica mediante *embeddings* y recuperación de fragmentos relevantes por similitud ( Fig. 4.2); (iii) reconocimiento de entidades y correferenciación con controles específicos para alias; (iv) extracción iterativa de relaciones con consolidación intermedia y deduplicación final; (v) generación de resúmenes (narrativo global y por personaje) reutilizables tanto como contexto de alto nivel como para tareas posteriores; y (vi) síntesis de salida (grafo de personajes y, opcionalmente, imágenes a partir de descripciones). Esta



junto con sus metadatos asociados, se almacenaron localmente para facilitar su reutilización y evitar la recomputación durante las distintas etapas del flujo.

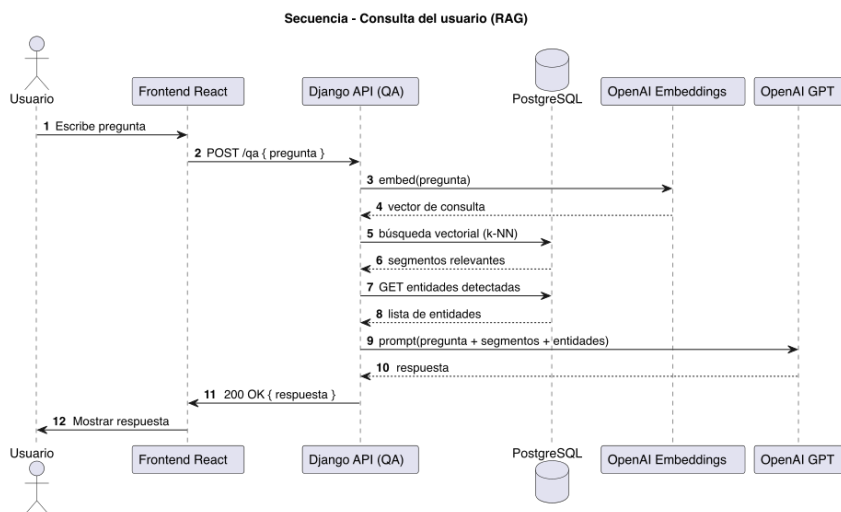


Figura 4.2: Secuencia de segmentación, indexación por *embeddings* y recuperación por similitud para RAG (consulta → *embedding* → vecinos más cercanos por coseno → contexto).

## Crterios de diseño y *trade-offs*

Como se mencionó anteriormente en 3.5.2, se observó que `text-embedding-3-small` ofreció mejor correlación semántica en el dominio narrativo específico y mayor compatibilidad con los LLM empleados, frente a `USE`.

**Métrica de similitud.** Se adoptó distancia coseno por su estabilidad con vectores normalizados; alternativas (p. ej., euclídea) no mostraron ventajas.

**Selección de *k*.** Valores altos incrementan cobertura con riesgo de ruido; valores bajos mejoran precisión pero pueden omitir evidencia. Se empleó un *k* intermedio (ajustado por longitud de obra y tarea).

**Tamaño de fragmento.** Ventanas pequeñas favorecen precisión local pero pierden relaciones implícitas distribuidas; ventanas grandes incrementan costo y riesgo de *lost in the middle*. Se adoptó un rango intermedio, ajustado empíricamente por obra.

**Correferencia.** Se decidió utilizar `spacy-experimental` y LLM para implementar la correferencia y comparar resultados.

## 4.2. Reconocimiento de entidades y correferencia

Se utilizó SpaCy (`en_core_web_trf`) para identificar entidades de tipo PERSON, aplicando reglas de normalización (tratamientos, compuestos, duplicados) y un módulo de correferencia. Con el objetivo de evaluar impacto y viabilidad, se implementaron dos variantes de correferenciación y se compararon experimentalmente:

- **Correferencia basada en *spaCy-experimental*.** Fue la primera opción integrada en el flujo por su disponibilidad e integración directa con SpaCy. Ofrece una solución liviana y estable, aunque sin capacidad de incorporar reglas de reconocimiento de alias en tiempo de ejecución.
- **Correferencia vía LLM (genérica y *alias-aware*).** Se incorporaron dos variantes con LLM:
  - (i) *genérica*, basada en una *prompt* de etiquetado de menciones que devuelve el texto original con marcas `[CHARACTER_ID] . . . [/CHARACTER_ID]` (Listings A.26–A.28), adaptada de (Amalvy y cols., 2024b); y
  - (ii) *alias-aware*, que permite inyectar reglas explícitas de alias en la *prompt*. Estas reglas incluyen inclusión de honoríficos y títulos, mapeos de equivalencias, preferencia por nombre completo, manejo de iniciales, y normalización tipográfica, con la hipótesis de mejorar la unificación de menciones (Listings A.29–A.30).

En términos de uso en el flujo, ambas variantes se aplican antes de la recuperación por *embeddings* y del reconocimiento de entidades con nombre. La variante con LLM se incorporó específicamente para evaluar el beneficio de tener conocimiento sobre las reglas de detección de alias frente a *spaCy-experimental*.

La correferencia con *spaCy-experimental* mostró mayor estabilidad y determinismo (variabilidad reducida entre corridas y menor costo/latencia), mientras que la alternativa con LLM ofreció mayor control por reglas y mejor capacidad para capturar alias no triviales, a costa de sensibilidad a la *prompt* y variabilidad en casos límite, además de un costo superior. La variante LLM tendió a reducir duplicaciones residuales (mejorando precisión en la unificación) pero puede introducir decisiones de fusión discutibles si la evidencia es ambigua.

De manera complementaria, la consolidación final basada en LLM, orientada a la unificación de personajes y deduplicación de relaciones con apoyo en resúmenes de identidad/rol, se ejecuta al final del módulo de extracción de redes y se describe en 4.5.5.

## 4.3. Generación de resúmenes

Se implementó un esquema iterativo por segmentos con dos líneas principales: (i) resumen narrativo global y (ii) resumen por personaje orientado a

descripción física. En ambos casos, el LLM recibe como contexto auxiliar el resumen acumulado y/o la lista de entidades, pero con instrucciones explícitas para resumir exclusivamente el fragmento de texto presentado en cada iteración. Luego, se utilizó una etapa de refinamiento de la salida utilizando nuevamente un LLM, basándonos en lo visto en 2.2.1. El flujo general de ambas variantes se ilustra en la Fig. 4.3.

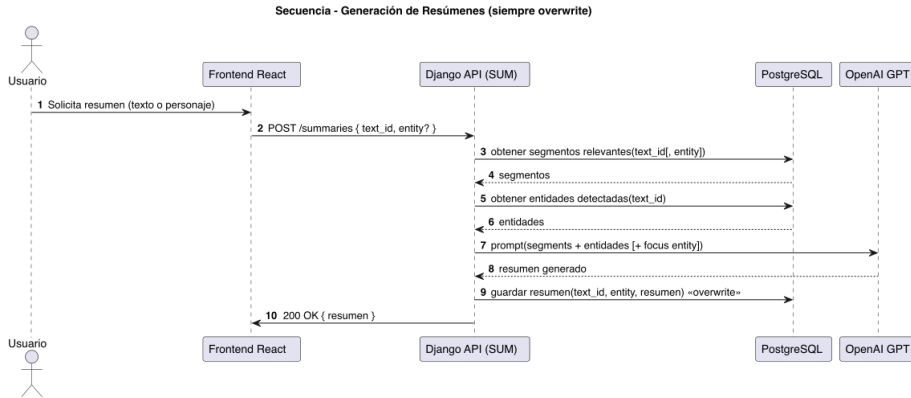


Figura 4.3: Flujo iterativo de resumen

El procedimiento para el resumen narrativo global se rige por el *system prompt* del apéndice A.13 y la plantilla de usuario del apéndice A.14. En cada iteración  $t$ , se entregan: (1) el resumen acumulado  $R_{t-1}$ , (2) la lista de entidades del texto completo y (3) el fragmento  $S_t$ . El *system prompt* instruye que ni el listado de entidades ni el resumen acumulado deben repetirse en la salida: su propósito es únicamente facilitar la tarea de síntesis del fragmento (punto 3). El resultado es un aporte local que, por agregación sucesiva, construye el resumen narrativo global (véase panel (a) de la Fig. 4.3).

Listing 4.1: Núcleo del system prompt para resumen por fragmentos (iterativo)

```
You summarize only the CURRENT fragment S_t.
Do NOT restate the accumulated summary or the entity list.
Write 2-4 sentences, factual and compact, capturing plot events and
entities mentioned.
If the fragment is non-informative for plot, return "NO SUMMARY".
Output: plain text.
```

Listing 4.2: Plantilla de usuario (insumos por iteración)

```
Inputs:
- Accumulated summary R_{t-1}: <texto breve>
- Global entity list (PERSON): ["Sherlock Holmes","John Watson",...]
- Current fragment S_t: <TEXTO_DEL_FRAGMENTO>
Task: Summarize S_t only. Do not copy R_{t-1} or the entity list.
```

Tras completar la pasada iterativa, se contemplan dos refinamientos opcionales, según el uso posterior:

- Enfoque “portada de libro”: Se aplica el *system prompt* del apéndice A.15, que reescribe el texto para enfatizar temas, atmósfera y elementos visuales relevantes a una portada, excluyendo detalles no visuales y manteniendo brevedad.

Listing 4.3: Refinamiento “portada de libro” (fragmento)

```
Rewrite the story overview as a book-cover blurb:  
- highlight mood, setting, key visual motifs (no spoilers)  
- 60-90 words, evocative but concise  
- avoid proper-noun overload; aim for imagery
```

- Resumen narrativo conciso para contexto: Se aplica el *system prompt* del apéndice A.16, que prioriza argumento, personajes principales, relaciones clave y contexto, produciendo una versión breve y coherente para ser reutilizada como contexto de alto nivel en otros módulos.

Listing 4.4: Refinamiento a resumen conciso para contexto (fragmento)

```
Condense to 4-6 sentences:  
- main premise, central conflict, key characters  
- keep explicit causal links; avoid conjecture  
- no quotes; neutral tone
```

Para la extracción de descripciones físicas se emplean el *system prompt* del apéndice A.17 y la plantilla de usuario del apéndice A.18. El LLM recibe: (1) el nombre del personaje objetivo, (2) un fragmento y (3) la lista de entidades del texto completo (para desambiguación). Se le instruye a devolver un párrafo conciso con rasgos físicos, indumentaria y, sólo cuando sea visualmente pertinente, detalles del rol. Si en el fragmento no existe descripción física, debe responder literalmente NO DESCRIPTION. El encadenamiento iterativo y el refinamiento posterior de estas descripciones se muestra en el panel (b) de la Fig. 4.3.

Listing 4.5: Descripción física por personaje (fragmento de system prompt)

```
TARGET: <PERSONAJE_OBJETIVO>. From the current fragment only, extract  
visual traits:  
physique, face, hair, clothes, distinctive items. If none present, return  
"NO DESCRIPTION".  
Output: 1 short paragraph, visual details only.
```

Listing 4.6: Entrada de usuario para descripción física (por fragmento)

```
Inputs:  
- Target entity: "Sherlock Holmes"
```

```
- Global entity list: ["Sherlock Holmes","John Watson",...]
- Current fragment: <TEXTO>
Task: Physical appearance only. If absent, "NO DESCRIPTION".
```

**Postprocesamiento para texto-a-imagen.** Las descripciones físicas se refinan con el *system prompt* del apéndice A.19, que elimina redundancias, resuelve contradicciones, restringe a que la información sea visualmente realizable y la reescribe para una mejor compatibilidad con modelos *text-to-image*. El resultado es el insumo directo del módulo de imágenes (4.4).

Listing 4.7: Postproceso para texto-a-imagen (fragmento)

```
Normalize to a single coherent physical description of the target:
- remove contradictions and duplicates
- keep only visualizable facts; avoid backstory
- produce a single prompt-like paragraph suitable for text-to-image
```

**Vínculo con alias y consolidación final.** El resumen narrativo global refinado (A.16) se reutiliza como contexto de alto nivel en la etapa de consolidación final de entidades y relaciones (véase 4.5.5). Asimismo, debido a que el resumen por personaje descrito aquí está orientado a describir la apariencia de los personajes, se introdujo adicionalmente una función específica para generar resúmenes por personaje orientados a identidad/rol utilizada solamente por el módulo de deduplicación (no cubiertos por las promps A.17–A.19).

Listing 4.8: Resumen por personaje orientado a identidad/rol (fragmento)

```
TARGET: <PERSONAJE>. Summarize identity/role signals from CURRENT
fragment:
occupation, social role, stable aliases, titles, relationships explicitly
stated.
No plot events; 2-3 sentences max. If nothing present, "NO DESCRIPTION".
```

Estos resúmenes se utilizan para apoyar la unificación de alias y la deduplicación de relaciones. Las plantillas de *prompt* empleadas para esta función (extracción iterativa y refinamiento) se detallan en el apéndice (Véase A.20–A.23).

## 4.4. Generación de imágenes (FLUX DEV + ComfyUI)

Se integró *FLUX DEV* (Artificial Analysis, 2025) con *ComfyUI* (comfy-anonymous, 2023) para la generación de retratos y portadas a partir de los resúmenes. Esta decisión se tomo en contraste con otros modelos *text to image*, debido a que este incorpora un LLM para un mejor alineamiento contra aquellos que utilizan representaciones conjuntas del texto y la imagen, como es

mencionado en la sección 2.5. Se observó mayor fidelidad en retratos; las portadas presentaron variabilidad superior y requieren *prompts* más específicos. El encadenamiento de entradas (resúmenes postprocesados) y salidas (artefactos visuales) se esquematiza en la Fig. 4.4.

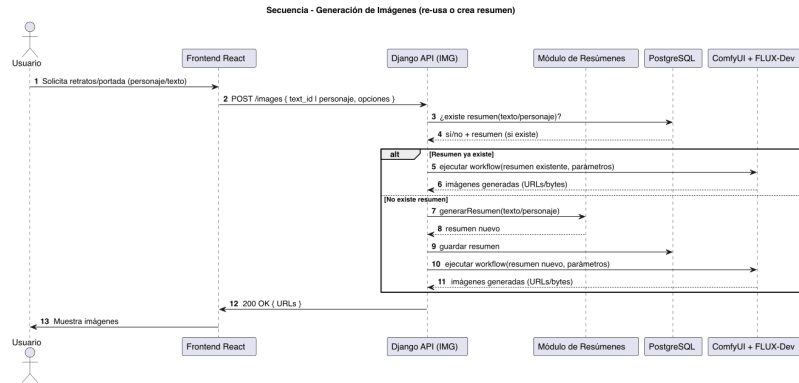


Figura 4.4: Integración de resúmenes postprocesados con el subsistema de generación de imágenes.

## 4.5. Extracción de redes de personajes

Los módulos de resúmenes (4.3) y de generación de imágenes (4.4) son independientes del núcleo de extracción, pero comparten insumos (segmentación, entidades) y reutilizan artefactos (p. ej., el resumen narrativo global como contexto de consolidación). La consolidación final de entidades y relaciones (unificación/deduplicación) se ejecuta dentro de este módulo, inmediatamente antes de la salida (4.5.5).

### 4.5.1. Planteo y estrategia

El modelo opera sobre segmentos recuperados para detectar relaciones entre personajes, contemplando vínculos implícitos y su posible evolución temporal (como se ejemplifica en 3.1). Para maximizar cobertura sin incrementar en exceso el costo monetario proveniente del uso de los modelos de lenguaje externos, se efectuaron múltiples pasadas y una deduplicación posterior, en línea con estrategias de auto-consistencia (X. Wang y cols., 2022) (Véase también 2.3.2). El encadenamiento de pasadas y la consolidación subsecuente se sintetizan en la Fig. 4.5.

### 4.5.2. Modelos

Se trabajó con `gpt-3.5-turbo` y, posteriormente, con `gpt-4o-mini` por su equilibrio costo monetario – precisión. *Llama 2* se descartó para este módulo

dado lo mencionado previamente (véase 3.5).

### 4.5.3. Variantes de *prompting*

Evaluamos dos formas de inyectar ejemplos (*in-context*): (i) Shots in System (ejemplos incluidos en el *system prompt*) y (ii) Shots in Chat (ejemplos incluidos junto con la instrucción principal, normalmente como mensajes de usuario o como mensajes de contexto.). Adicionalmente, analizamos el efecto de dos señales de apoyo del pipeline (no son variantes del *prompt*): (iii) lista de entidades provista al modelo y (iv) correferencia previa. En nuestros resultados, *Shots in System* tendió a ofrecer un equilibrio más sólido entre precisión y cobertura (mejor  $F_1$ ), mientras que *Shots in Chat* alcanzó precisiones más altas en algunos escenarios. La lista de entidades ayudó a reducir falsos positivos en las mejores configuraciones, y la correferencia previa mostró efectos mixtos según su implementación. La Fig. 4.5 resume el flujo completo en el que se insertan estas decisiones.

### 4.5.4. Diseño de *prompts*

El diseño de los *prompts* evolucionó desde plantillas iniciales hacia estructuras más guiadas, incorporando prácticas provenientes de trabajos recientes en extracción de relaciones con LLMs, particularmente el uso combinado de *system prompts*, *user prompts*, enfoques *few-shot* y técnicas inspiradas en *Chain-of-Thought* (J. Wei y cols., 2022; Wan y cols., 2023). Esta evolución incluyó la definición operativa de los tipos de relación, el uso de ejemplos en *Shots in System* y *Shots in Chat*, y la incorporación progresiva de criterios de trazabilidad y control de salida.

#### Role prompting

Se explicitó un rol experto para el modelo (por ejemplo, “analista especializado en extracción de relaciones entre personajes de textos narrativos”) con el objetivo de orientar la lectura, reducir ambigüedad y estabilizar el criterio de decisión. Este marco de rol funcionó como ancla interpretativa ante variaciones estilísticas entre fragmentos y permitió obtener relaciones más coherentes y menos ruidosas.

#### Supporting text

Para cada relación identificada se exigió una justificación breve basada en un fragmento del texto fuente. Esta práctica, inspirada en mecanismos de *rationale extraction*, aporta trazabilidad, posibilita auditoría manual y facilita el filtrado de casos ruidosos o inconsistentes. Además, obligar al modelo a fundamentar cada enlace demostró reducir alucinaciones y mejorar precisión en configuraciones intermedias.

## Salida estructurada

Se adoptó un formato de salida estable en JSON, con campos explícitos para `Personaje1`, `Personaje2`, tipo de relación, `supporting_text` y ubicación o segmento textual. Esta estandarización reduce la variabilidad en las respuestas del modelo, simplifica las etapas de posprocesamiento y permite convertir directamente las salidas en `dataframes` de `pandas`, facilitando su manipulación sistemática y la integración con las etapas posteriores de evaluación automática.

Esta progresión, desde plantillas libres hasta esquemas guiados con evidencia textual y estructura fija, se documenta en la sección 3.5.3 y en los apéndices correspondientes (A.1.1, A.1.2, A.1.3, A.1.4).

### 4.5.5. Consolidación final basada en LLM (deduplicación de personajes y relaciones)

Previo a la entrega de resultados, se incorpora una etapa de consolidación final que solicita al LLM la unificación de personajes y la deduplicación de relaciones. Para ello se introduce una función específica que genera un resumen orientado a identidad/rol del personaje (distinto del resumen de rasgos físicos, 4.3), a fin de proporcionar evidencia semántica suficiente para decidir fusiones con criterio narrativo. La *prompt* final recibe como entrada: (i) un resumen del texto, (ii) el resumen por personaje (identidad/rol) y (iii) la lista de relaciones extraídas; con esta información, el LLM identifica equivalencias entre personajes (alias) y relaciones redundantes, y devuelve un mapeo de fusión de entidades junto con el grafo depurado. El resumen por personaje (identidad/rol) utilizado como insumo se genera con las prompts del apéndice (Véase A.20–A.23).

Este procedimiento se articula con el acercamiento planteado por (Amalvy y cols., 2024b), donde se emplean LLM en la etapa de resolución de correferencias en conjunto con el reconocimiento de entidades nombradas para normalizar menciones antes de construir la red. En nuestro caso, la correferencia *alias-aware* (4.2) aporta señales para homogeneizar entidades, y la presente etapa de consolidación las complementa con criterios narrativos a partir de resúmenes de identidad/rol. Las plantillas de *prompt* empleadas en esta etapa se incluyen en los apéndices A.31–A.32, basadas en pautas del enfoque CREFT (Chun y cols., 2025).

### 4.5.6. Limitaciones

Una primera limitación identificada en la extracción automática de relaciones es la dificultad para reconstruir vínculos cuya evidencia se encuentra distribuida a lo largo del texto. Muchas relaciones literarias no se expresan en un único pasaje, sino que emergen de la combinación de fragmentos dispersos: antecedentes, interacciones centrales y resoluciones posteriores. Cuando el sistema sólo recupera ventanas parciales, el modelo carece de la información necesaria para reconstruir la relación completa.

Esto se observa con claridad en *A Scandal in Bohemia*. Allí, el conflicto entre el Rey de Bohemia e Irene Adler sólo puede comprenderse combinando tres zonas del cuento: (i) el planteo inicial, donde el rey contrata a Holmes para recuperar una fotografía comprometedora; (ii) los episodios centrales que explican la naturaleza del conflicto y la relevancia de la fotografía; y (iii) el desenlace, donde Adler revela que ha decidido marcharse y casarse con otro hombre. Si el sistema recupera únicamente el encargo inicial o solo el final, la relación queda incompleta o directamente inaccesible para el modelo. Algo similar ocurre en *The Adventure of the Dancing Men*: la relación PERPETRATOR-VICTIM depende de integrar pistas cifradas (una serie de dibujos infantiles), la llegada del responsable al hogar de la víctima y el enfrentamiento final. Como estas piezas están separadas a gran distancia narrativa, fragmentos aislados no permiten reconstruir la cadena causal. Un caso aún más marcado aparece en *A Study in Scarlet*, donde la motivación del antagonista se narra en una segunda parte casi independiente de la primera. Si la recuperación no incorpora fragmentos de ambas secciones, el modelo sólo ve efectos sin causas o causas sin efectos, lo que impide inferir correctamente las relaciones entre personajes.

Una segunda limitación corresponde a los vínculos implícitos o ambiguos, que requieren señales globales del texto para interpretarse de forma adecuada. En ausencia de información suficiente, el LLM tiende a completar los vacíos con inferencias no justificadas, lo que puede llevar a alucinaciones (W. X. Zhao y cols., 2023).

Esto ocurre, por ejemplo, cuando el modelo recibe fragmentos parciales de *A Scandal in Bohemia*. Si el pasaje recuperado muestra que Holmes admira la inteligencia de Adler pero no incluye la aclaración final del narrador, según la cual nunca existió un vínculo romántico, el modelo puede inferir erróneamente una relación sentimental. En *The Man with the Twisted Lip*, el conflicto central gira en torno a un hombre que oculta su identidad bajo un disfraz y lleva una doble vida. Cuando el sistema recupera solo partes donde aparece la identidad secundaria, el modelo puede tratarla como un personaje separado y generar relaciones incompatibles entre ambas “personas”. De manera análoga, en *A Case of Identity*, un personaje se hace pasar por otro para manipular a la protagonista; sin el fragmento que revela el engaño, el modelo interpreta ambos nombres como individuos distintos, produciendo relaciones duplicadas. Finalmente, en *The Adventure of the Speckled Band*, la identificación del responsable del crimen depende casi por completo del desenlace: si el sistema recupera únicamente las etapas preliminares de la investigación, el modelo puede proponer relaciones causales incorrectas o insuficientemente fundamentadas, pues carece de la evidencia que confirma el método y la autoría.

En conjunto, estas limitaciones muestran que la extracción robusta de relaciones depende críticamente de la disponibilidad de contexto narrativo suficiente. Tanto la incompletitud en la recuperación de fragmentos como la presencia de identidades dobles, alias o vínculos implícitos pueden inducir vacíos que los modelos llenan con inferencias espurias, afectando la precisión y la coherencia de las relaciones obtenidas.

### 4.5.7. Resultado

El módulo final combina recuperación por *embeddings*, NER con SpaCy y correferencia, con *gpt-4o-mini* para inferencia. Se identifican los tipos definidos en el corpus<sup>1</sup> (Personales, profesionales, de sospecha y de víctima y victimario), generando una red interpretable. El flujo operativo y la fase de deduplicación previa a la visualización del grafo se muestran en la Fig. 4.5.

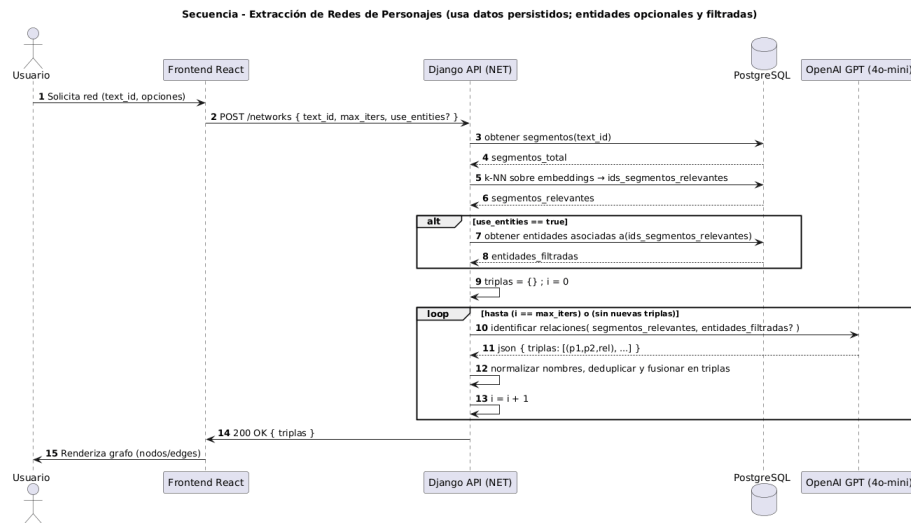


Figura 4.5: Flujo de extracción de relaciones por pasadas, consolidación intermedia y deduplicación final antes del grafo.

### 4.5.8. Enfoques alternativos explorados tras la solución base

Una vez establecida la solución base para extracción de relaciones (recuperación por *embeddings* + inferencia con LLM sobre segmentos relevantes), se evaluaron dos paradigmas recientes como líneas alternativas/ complementarias: ChatIE (X. Wei y cols., 2023) y SumAsk (G. Li y cols., 2023). Aquí se documenta exclusivamente lo realizado (los fundamentos se presentan en el Cap. 2), y los resultados comparativos se remiten a la sección 5.4.

- **ChatIE.** Se intentó adaptar la descomposición en subtareas tipo QA multi-turno al dominio narrativo. La elevada cantidad de relaciones implícita de las relaciones y las grandes distancias entre evidencias dificultaron su traslado directo, por lo que no se integró al flujo final. Se dejó como alternativa para realizar pruebas y comparar resultados.

<sup>1</sup>El corpus construido para este proyecto incluye anotaciones de relaciones interpersonales en textos de Sherlock Holmes, con categorías específicas para análisis narrativo. Disponible en: <https://gitlab.fing.edu.uy/proyecto-final2/corpus-sherlock-holmes>

- **SumAsk.** Se implementó una versión completa y se realizaron pruebas en configuración de clasificación de relaciones por par (asumiendo entidades dadas). Se analizan sus resultados y *trade-offs* frente al enfoque base en la sección 5.4.

## 4.6. Interfaz del sistema: diseño, componentes y funcionalidades

Esta sección describe la interfaz desarrollada para el prototipo desde una perspectiva de ingeniería: objetivos de diseño, arquitectura tecnológica, componentes principales y flujo de datos entre el *frontend* y el *backend*. El foco se sitúa en qué se implementó y cómo.

### 4.6.1. Objetivos de diseño

La interfaz se concibió con tres objetivos: (i) consolidar en un único entorno las capacidades del sistema (carga de textos, generación de resúmenes, extracción y visualización de redes, consultas contextuales y generación de imágenes), (ii) exponer artefactos intermedios relevantes para trazabilidad (segmentos, entidades, resúmenes y redes de personajes), y (iii) habilitar exploración interactiva con baja fricción y costos de cómputo acotados. Se priorizó un diseño minimalista, con jerarquía visual clara y separación de responsabilidades entre vistas.

### 4.6.2. Arquitectura tecnológica

La aplicación se implementó como solución web con Django (backend) y React (frontend), sobre PostgreSQL como base de datos. Esta elección facilita la integración con el ecosistema Python del preprocesamiento (NER, segmentación, *embeddings*) y ofrece componentes reutilizables en el cliente para visualización y navegación.

#### Backend

El backend expone endpoints REST para: registro de textos, consulta de metadatos (panel de administrador), disparo de tareas (resúmenes, red de personajes, imágenes) y recuperación de resultados. El modelo de datos persiste:

- Documentos: identificador, nombre y estado de procesamiento.
- Segmentos: texto y *embedding* asociado.
- Entidades: nombre y segmentos asociados.
- Resúmenes: narrativo global y por personaje (apariencia e identidad/rol, 4.3).

Se empleó el ORM de Django para operaciones CRUD de PostgreSQL para metadatos flexibles (listas de entidades por segmento, resúmenes, etc.). La autenticación/permiso se mantuvo simple, con miras a un prototipo académico. Además, Django provee un panel de administración desde el cual se puede acceder y gestionar todos los datos guardados en la base de datos.

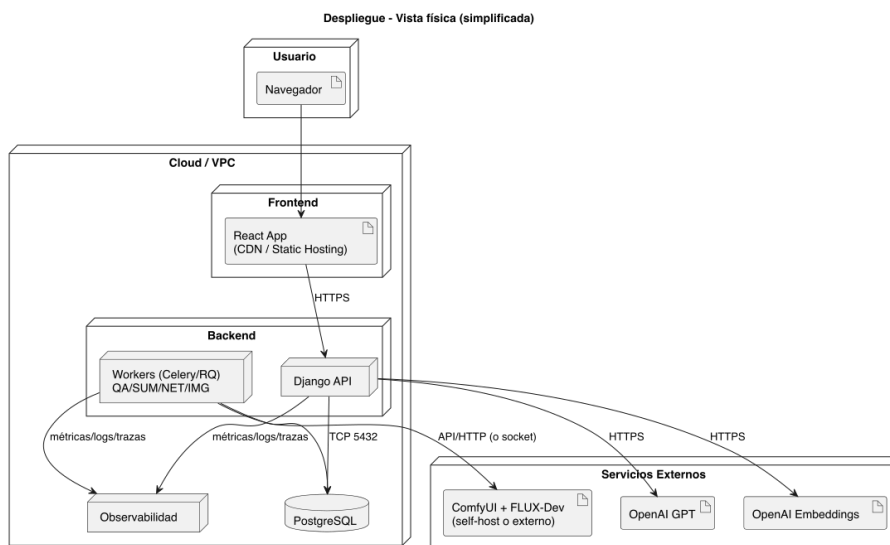


Figura 4.6: Arquitectura de despliegue: servicios de backend (API), base de datos y componentes externos.

## Frontend

El frontend en React organiza la UI en vistas autocontenidas y componentes reutilizables: tablas de documentos, paneles de control, visualización de grafos y tarjetas de personaje. El estado local (filtros, selección de documento/personaje) se maneja con *state hooks* y se evita el acoplamiento entre vistas compartiendo únicamente identificadores y llaves de caché. Las llamadas al backend se encapsulan en servicios dedicados (fetch con cancelación y reintentos básicos).

### 4.6.3. Vistas y funcionalidades

#### Listado de textos

La vista inicial presenta un inventario de los documentos registrados en el sistema, ofreciendo acciones de inspección y administración que funcionan como punto de entrada a los distintos flujos de procesamiento. Esta vista permite una navegación directa hacia el análisis de cada texto y constituye el eje organizador de la interacción con la interfaz.

El alta de nuevos documentos se realiza mediante un diálogo modal que solicita el nombre asignado y el archivo en formato de texto plano. Antes de enviarse al backend, la interfaz valida tanto el tamaño como el formato del archivo, asegurando la integridad del proceso.

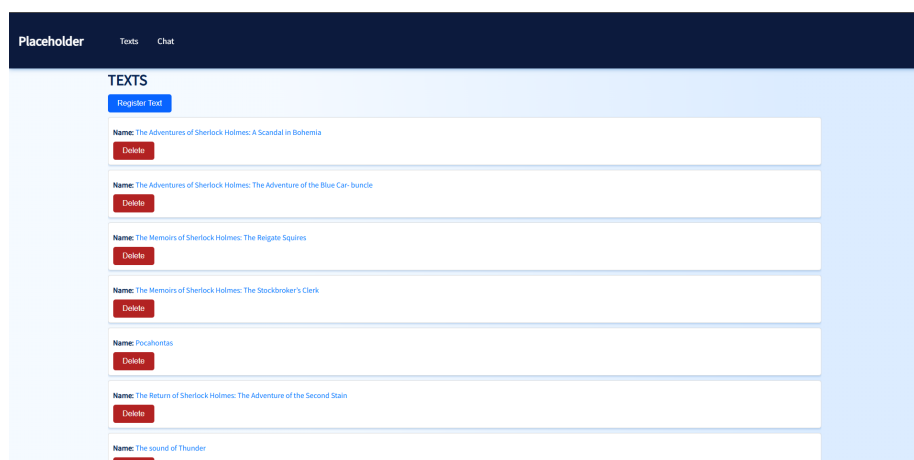


Figura 4.7: Vista de la sección “Listado de textos”, que presenta los documentos registrados y las acciones disponibles para cada uno.

### Vista de texto (análisis individual)

En la vista de documento se centralizan los artefactos de análisis: generación de resúmenes, portada y grafo de relaciones. La interfaz refleja el flujo de funcionamiento del sistema (preprocesamiento, recuperación y LLM), pero expone únicamente los resultados consolidados y trazas esenciales (Fig. 4.9).

**Resúmenes.** La vista permite disparar la generación y refinamiento del resumen narrativo global de acuerdo con la arquitectura descrita en 4.3; el resultado queda disponible en la misma vista (Fig. 4.9).

**Imagen de portada.** A partir de las descripciones postprocesadas (4.3), se habilita la solicitud de imágenes de portada (4.4); el artefacto generado se visualiza en la vista de texto (Fig. 4.9).

### Visualización de la red de personajes

El grafo presenta nodos (personajes) y aristas etiquetadas (tipos de relación definidas en el corpus). Se incluyeron interacciones básicas, como resaltado de nodo y acceso a la vista de cada personaje, tal como se observa en la Fig. 4.10.

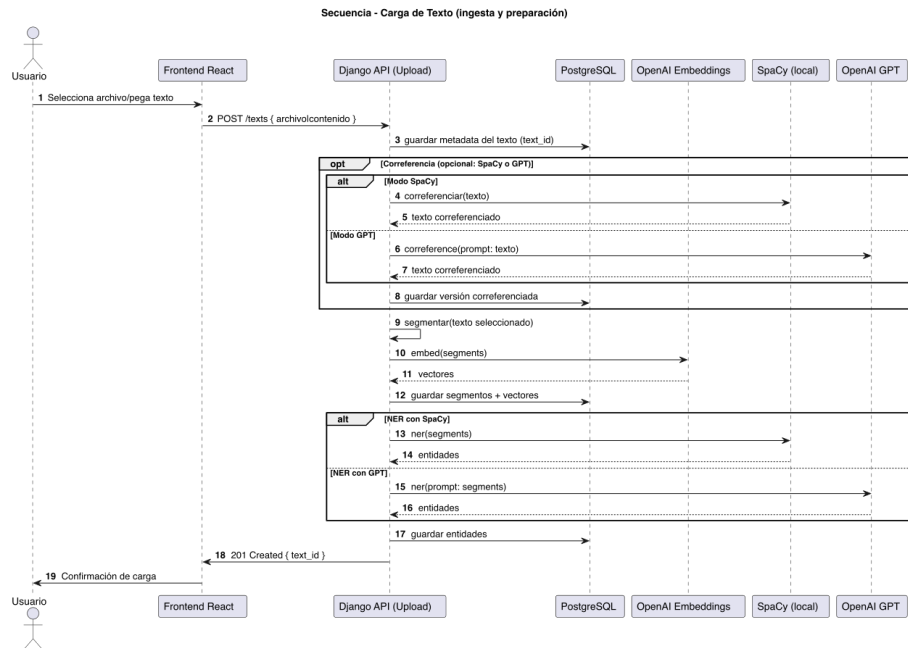


Figura 4.8: Secuencia de la operación “carga de texto” desde la interfaz: UI → API → persistencia y estados.

## Vista de personaje

La vista individual de personaje consolida: (i) resumen específico (apariciencia) y (ii) artefactos visuales (retrato). Esto facilita la revisión de consistencia entre texto e imagen generada (Fig. 4.11).

## Chat contextual

Se incluyó una interfaz de chat para consultas *ad hoc* sobre un documento seleccionado. La interacción utiliza recuperación por *embeddings* (4.1) y consolidación de fragmentos relevantes antes de la inferencia con LLM. En la Fig. 4.12 se puede observar un ejemplo de la interacción con el sistema.

### 4.6.4. Decisiones de implementación

Se adoptaron decisiones orientadas a viabilidad, trazabilidad y coste:

- **Acoplamiento laxo entre UI y servicios.** El frontend no codifica reglas de negocio; invoca endpoints idempotentes con contratos estables y deja el control de versiones del flujo al backend.
- **Interactividad razonable.** La UI presenta estados claros de ejecución

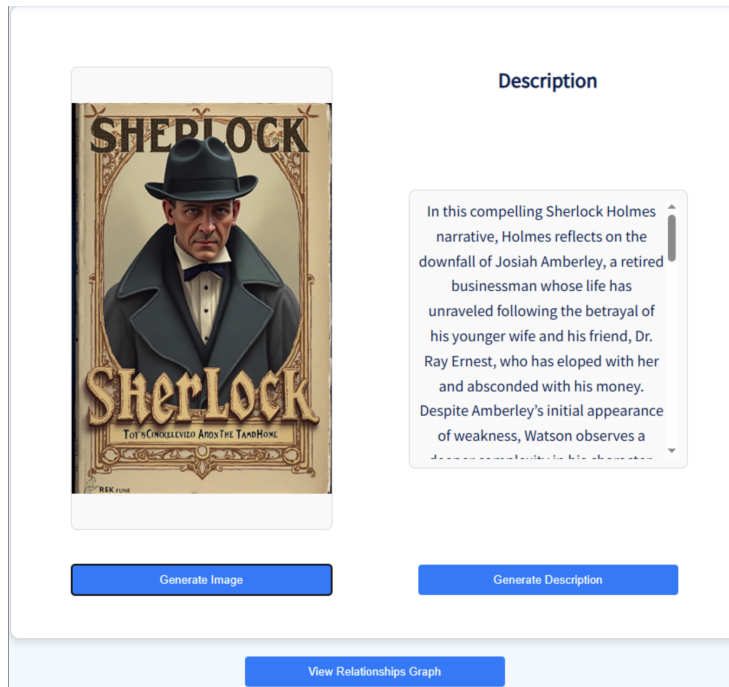


Figura 4.9: Vista de texto: acceso a resúmenes, portada y grafo de relaciones.

(pendiente, en curso, listo) y evita bloqueos; se utilizan indicaciones optimistas con *polling* discreto para sincronización.

#### 4.6.5. Flujo de datos e integración con módulos

Las vistas consumen resultados del flujo descrito en capítulos anteriores:

1. Preprocesamiento y recuperación (4.1): segmentación y *embeddings*; la UI sólo visualiza resúmenes/relaciones consolidadas y expone resultados (Fig. 4.9).
2. Resúmenes (4.3): narrativo global (para contexto de alto nivel) y por personaje (aparición e identidad/rol). La UI ofrece disparadores y muestra versiones refinadas, y solo disponibiliza el resumen por aparición (Fig. 4.9).
3. Alias y consolidación (4.2–4.5.5): la UI refleja el grafo ya deduplicado, luego del chequeo de *alias* (Fig. 4.10).
4. Imágenes (4.4): se listan artefactos generados (retratos/portadas) junto a la descripción usada (Fig. 4.11).

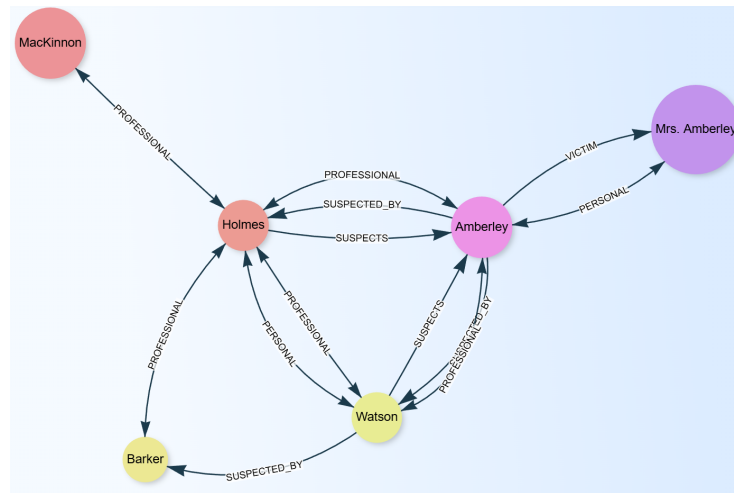


Figura 4.10: Vista de red de personajes con aristas tipadas y acceso a evidencia.

#### 4.6.6. Limitaciones y trabajo futuro de la interfaz

La interfaz presenta dos limitaciones principales que orientan líneas claras de trabajo futuro. En primer lugar, la escalabilidad visual: en textos con un número elevado de entidades, la red tiende a volverse densa y difícil de interpretar, lo que afecta la legibilidad de los patrones relacionales. Una mejora natural consiste en incorporar técnicas de agrupamiento por comunidades, así como la posibilidad de explorar vistas parciales del grafo (por subgrafo o por escena), facilitando un análisis más segmentado y manejable (Fig. 4.10).

En segundo lugar, la interfaz conserva únicamente el último estado consolidado de la extracción, lo que limita la trazabilidad y la comparación entre corridas. Resultaría valioso habilitar la visualización de *snapshots* correspondientes a diferentes pasadas o configuraciones, permitiendo estudiar la estabilidad del sistema, identificar variaciones y analizar la evolución de los resultados.

En conjunto, estas mejoras apuntan a fortalecer la utilidad analítica de la interfaz, ampliando su capacidad para manejar casos más complejos y ofreciendo una lectura más controlada y transparente de los resultados.

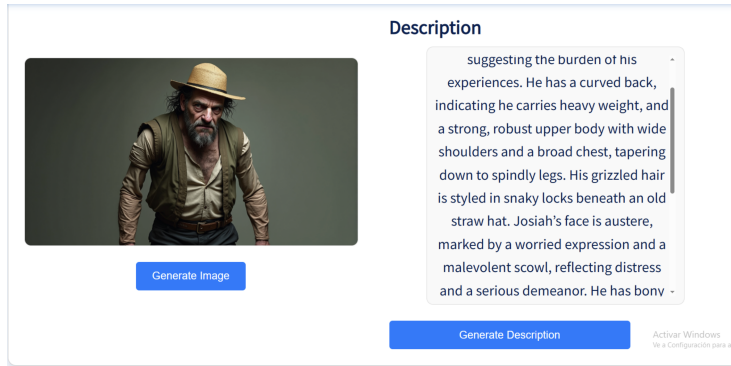


Figura 4.11: Vista de personaje: resúmenes específicos, relaciones incidentes y retrato.

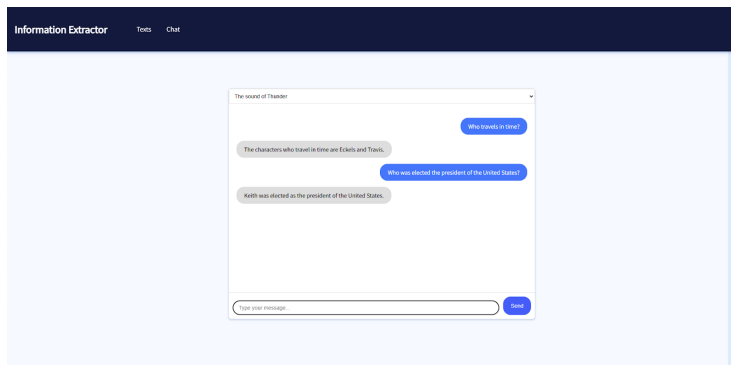


Figura 4.12: Historial de mensajes y trazas de contexto empleadas.



## Capítulo 5

# Experimentación

Este capítulo presenta el diseño y los resultados de la evaluación empírica del sistema. El objetivo es caracterizar, con evidencia cuantitativa y trazable, cómo distintas decisiones de orquestación, segmentación y recuperación, variantes de *prompting*, provisión de lista de entidades, correferencia, número de corridas y algoritmo de extracción, impactan en la calidad de las relaciones recuperadas. La experimentación se apoya en el corpus anotado construido para este trabajo y reporta métricas estándar (precisión, *recall* y  $F_1$ ), complementadas con consideraciones de costo (llamadas/tokens) cuando corresponde.

La evaluación se organiza en tres etapas: (i) una cuadrícula base que recorre combinaciones de hiperparámetros del pipeline *vanilla*; (ii) un subestudio controlado sobre correferencia basada en LLM y extracción de entidades en la configuración más prometedora; y (iii) experimentos adicionales con variantes de extracción y con postprocesamiento de consolidación (deduplicación), aplicado selectivamente por su costo. Todas las corridas se realizaron con una partición fija del corpus para favorecer reproducibilidad.

A continuación se describen los escenarios de evaluación y los hiperparámetros considerados.

### 5.1. Escenarios de evaluación

Para evaluar el desempeño de la aplicación, se utilizó el corpus construido en este proyecto, dividido en un conjunto de entrenamiento/validación y un conjunto de pruebas (vease [A.2.2](#)). Durante el desarrollo se trabajó con el primero para ajustar parámetros y descartar enfoques, mientras que el segundo se destinó a la evaluación final.

El análisis experimental se centró en el módulo de extracción de relaciones, núcleo del sistema, aunque también se evaluaron configuraciones asociadas al preprocesamiento y a la forma en que la información se presentaba al LLM. Para ello se definieron los siguientes hiperparámetros de control, cada uno con una función específica dentro del flujo:

- **Modelo de lenguaje empleado.** Se compararon distintas versiones de LLM (GPT-3.5-turbo y GPT-4o-mini). Este parámetro determina la capacidad de razonamiento y la calidad de las respuestas generadas, así como el costo computacional de las inferencias.
- **Módulo de correferencia.** Se probó activado y desactivado. Su objetivo es unificar menciones distintas de un mismo personaje (ej. “Holmes”, “Mr. Holmes”), lo que puede influir directamente en la coherencia del reconocimiento de relaciones.
- **Tipo de system prompt.** Se consideraron dos variantes: los ejemplos incluidos directamente en el prompt del sistema (*Shots in System*) y los ejemplos introducidos como parte de un diálogo previo que simula una conversación (*Shots in Chat*). La forma en que se estructuran las instrucciones iniciales puede afectar cómo el modelo interpreta la tarea de extracción (Brown y cols., 2020).
- **Lista de entidades.** En algunos escenarios se proporcionó al modelo la lista de personajes detectados en los segmentos, mientras que en otros debía inferirlos directamente del texto. Este parámetro controla el nivel de apoyo externo dado al LLM en el reconocimiento de las entidades relevantes.
- **Número de ejecuciones (runs).** Se configuró el sistema para ejecutar el algoritmo una o varias veces sobre el mismo texto, consolidando luego los resultados. Este parámetro define la estrategia de cobertura: cuantas más corridas se hagan, mayor es la probabilidad de eliminar relaciones generadas por alucinación del LLM.
- **Cantidad de segmentos enviados simultáneamente.** Dado que los textos fueron divididos en fragmentos, este hiperparámetro regula cuántos se entregan al LLM en cada consulta. Controla el balance entre la cobertura contextual y las limitaciones de ventana de tokens de los LLM.
- **Algoritmo de extracción.** Se comparó el enfoque original con las variantes de SumAsk y ChatIE. Este parámetro determina la lógica con la que se generan y consolidan las relaciones propuestas por el LLM.

En conjunto, estos hiperparámetros estructuran los escenarios de evaluación, ya que permiten variar el LLM, el contexto y la estrategia de inferencia, influyendo en la cobertura, precisión y costo computacional del sistema.

## 5.2. Partición del corpus y justificación

Con el objetivo de garantizar una evaluación no sesgada, se realizó una partición a nivel de cuento en dos subconjuntos: un conjunto reducido de desarrollo (entrenamiento/validación) de 10 cuentos y un conjunto de prueba con el resto

de los relatos. Como en este trabajo no se entrenan parámetros internos de los LLM (no hay *fine-tuning*), el conjunto de desarrollo se destinó exclusivamente al ajuste de hiperparámetros del flujo (cantidad de fragmentos,  $k$  de recuperación por *embeddings*, variantes de *prompt*, umbrales de consolidación, etc) y a verificar la estabilidad de la orquestación, reservando la mayor parte del material para una medición final independiente.

La partición se efectuó por relato completo, sin fragmentar cuentos entre subconjuntos, para evitar fugas de información entre desarrollo y prueba, por ejemplo, que alias o relaciones inferidas indirectamente a partir de un cuento reaparezcan en el conjunto de evaluación. Asimismo, se procuró una representación balanceada de los tipos de relación dentro del subconjunto de desarrollo, de forma de no sesgar el ajuste hacia un único patrón narrativo. La selección se fijó de manera determinística y se mantuvo invariable durante el estudio.

El listado explícito de los cuentos asignados al subconjunto de desarrollo se incluye en el apéndice (A.2.2); los restantes se destinaron al conjunto de prueba.

### 5.3. Metodología de evaluación

La evaluación experimental se diseñó para medir la capacidad del sistema de extraer correctamente relaciones entre personajes a partir de textos narrativos. Para ello, se trabajó con el conjunto de prueba definido, organizado en carpetas independientes cada una correspondiente a un cuento, que incluyen: (i) el texto original en formato plano, (ii) un archivo `Personajes.csv` con los nombres canónicos de los personajes y sus variantes o apodos, y (iii) un archivo `Relaciones.csv` que define el conjunto de relaciones anotadas manualmente y utilizadas como verdad de referencia.

A partir del texto de entrada, el sistema genera una lista de tripletas del tipo ( $Personaje_1, Personaje_2, Relación$ ). Antes de contrastar estas predicciones con el conjunto de verdad, se aplica un paso de normalización y consolidación orientado a eliminar duplicados. Este proceso agrupa variantes superficiales de los nombres detectados —como diferencias de mayúsculas, puntuación o títulos— y, de forma central, utiliza la lista de alias y denominaciones alternativas provista en el archivo `Personajes.csv` para unificar referencias a un mismo personaje. De este modo, se evita contabilizar múltiples veces una misma relación expresada con distintas denominaciones de los personajes involucrados.

Las relaciones predichas, una vez consolidadas, se comparan con las relaciones del archivo `Relaciones.csv`. Una predicción se considera correcta si coincide con una relación del conjunto de verdad tanto en el tipo de vínculo como en los personajes involucrados, admitiendo equivalencias entre nombres mediante el uso de alias y variantes. Esta comparación se realiza a nivel de cuento, respetando la anotación independiente de cada texto.

Sobre la base de esta correspondencia se calculan métricas estándar de extracción de relaciones (Jurafsky y Martin, 2008, 2025), utilizando conteos globales de verdaderos positivos, predicciones totales y relaciones reales. En particular, se reportan precisión, recall y  $F_1$ , donde la precisión mide la proporción de re-

laciones predichas que son correctas, el recall indica la fracción de relaciones reales que el sistema logra recuperar, y el micro  $F_1$  Score se emplea como métrica principal, ya que refleja el equilibrio entre precisión y cobertura y ofrece una visión más justa del rendimiento global del sistema al agregarse sobre todas las predicciones evaluadas. Este esquema de evaluación permite comparar de forma consistente las distintas configuraciones experimentales analizadas en el capítulo.

De manera complementaria, se tuvo en cuenta el costo computacional asociado a cada configuración, medido en términos de llamadas a la API y cantidad de *tokens* consumidos, aunque este aspecto no se reporta como métrica formal en los resultados.

## 5.4. Resultados experimentales

Los resultados evidencian un comportamiento claramente diferenciado según la combinación de hiperparámetros. La cuadrícula principal sin correferencia activada (en adelante, algoritmo *vanilla*) se resume en la Tabla 5.1; el impacto de activar correferencia se muestra en la Tabla 5.2; y la Tabla 5.3 condensa las configuraciones con mejor  $F_1$ . Además, se reporta un subestudio controlado de correferencia basada en LLM (plantillas genérica y *alias-aware*) y extracción de entidades (spaCy vs. LLM) en la Tabla 5.4, así como el postprocesamiento de consolidación (deduplicación) aplicado sobre las configuraciones más fuertes en la Tabla 5.5. Por último, se incluyen las variantes SumAsk (Tabla ??) y ChatIE (Tabla ??), evaluadas de forma acotada por su mayor costo computacional, y un muestreo de modelos recientes (Sec. 5.4) que ilustra el techo alcanzable bajo mayores costos.

**Resumen de hallazgos principales.** En términos globales, las mejores métricas se obtuvieron con correferencia desactivada, empleando Shots in System. En particular, el mejor  $F_1$  observado para el algoritmo vanilla sin postprocesamiento fue 0.28 (empate) con el módulo de correferencia desactivado, *Shots in System*, pasando la lista de entidades, tanto con 3 y 5 runs como con 12 (véanse Tabla 5.1 y Tabla 5.3). En ese mismo escenario se alcanzó también el mejor recall (0.25). Por su parte, Shots in Chat lograron las mejores precisiones (hasta 0.39) cuando el módulo de correferencia estaba desactivado y se enviaba la lista de entidades, con 8 runs, pero con  $F_1$  0.26, por debajo del máximo (Tabla 5.1). Esto sugiere que, aunque *Shots in Chat* puede elevar la precisión, *Shots in System* ofrece un mejor equilibrio general entre precisión y cobertura ( $F_1$ ). La activación del módulo de correferencia utilizando spacy experimental degrada sistemáticamente las métricas para casi todas las combinaciones (Tabla 5.2). Respecto a la lista de entidades, los mejores  $F_1$  (0.28) se obtuvieron pasando la lista de entidades y con *Shots in System*; sin la lista de entidades y con *Shots in System* también se observan resultados competitivos ( $F_1$  hasta 0.24 con 8 runs; Tabla 5.1). El número de corridas muestra rendimientos decrecientes: pasar de

**Tabla 5.1:** Efecto combinado de Shots, lista de entidades y número de corridas con `gpt-4o-mini` sin correferencia. Se reportan precisión, recall y  $F_1$ . Mejor valor global en **negrita**; segundo mejor subrayado.

Shots	Entidades	Runs	Precisión	Recall	$F_1$
Chat	Sí	3-5	<u>0.37-0.38</u>	0.20	<u>0.26</u>
		8	<b>0.39</b>	0.17	0.23
		12	0.32	0.15	0.21
	No	3	0.22	0.07	0.11
		5-8	0.30-0.33	0.09	0.14-0.15
		12	0.30	0.08	0.12
System	Sí	3-5	0.32	<b>0.25</b>	<b>0.28</b>
		8	0.34	0.21	<u>0.26</u>
		12	0.35	<u>0.23</u>	<b>0.28</b>
	No	3-5	0.29-0.30	0.16-0.18	0.21-0.22
		8-12	0.35-0.37	0.18	0.23-0.24

**Tabla 5.2:** Impacto de correferencia activada con `gpt-4o-mini`: combinación de Shots, entidades y corridas. Se marca el mejor valor global en **negrita** y el segundo mejor global subrayado.

Shots	Entidades	Runs	Precisión	Recall	$F_1$
Chat	Sí	3-5	0.12-0.14	0.04-0.05	0.06-0.08
		8-12	0.10-0.13	0.02-0.04	0.04-0.07
	No	3	0.14	<b>0.10</b>	<b>0.12</b>
		5-12	0.14-0.16	0.08- <u>0.09</u>	0.10- <b>0.12</b>
System	Sí	3-5	0.10-0.14	0.02-0.04	0.04-0.06
		8	<b>0.22</b>	0.04	0.07
		12	0.03	0.00	0.01
	No	3-5	0.12-0.14	0.05-0.08	0.07-0.10
		8	<u>0.18</u>	0.07	0.10
		12	0.12	0.04	0.06

**Tabla 5.3:** Mejores configuraciones ordenadas por  $F_1$ , con módulo de correferencias desactivado y enviando la lista de entidades. Se marca el mejor valor global en **negrita** y el segundo mejor global subrayado.

Shots	Runs	Precisión	Recall	$F_1$
System	3	0.32	<b>0.25</b>	<b>0.28</b>
	5	0.32	<b>0.25</b>	<b>0.28</b>
	12	0.35	<u>0.23</u>	<b>0.28</b>
Chat	3	<u>0.37</u>	0.20	<u>0.26</u>
	5	<b>0.38</b>	0.20	<u>0.26</u>

3 a 5–8 corridas estabiliza y mejora levemente  $F_1$ , pero más allá los incrementos son marginales.

#### Variantes de correferencia y NER con LLM (subestudio controlado).

Como el análisis global ubicó el mejor desempeño con el módulo de correferencia desactivado, ejecutamos una serie adicional acotada sobre la combinación más prometedora (*Shots in System* con lista de entidades), con el fin de aislar el efecto de: (i) la plantilla de correferencia basada en LLM ,genérica vs. *alias-aware*<sup>1</sup>, y (ii) la extracción de entidades ,*spaCy* vs. extracción de entidades con LLM, variando  $runs \in \{3, 5\}$ . Se corrieron menos tests a propósito para concentrar costos en el escenario más relevante.

En este marco, el mejor  $F_1$  fue 0.26 con correferencia genérica y *spaCy* para extracción de entidades y con 5 runs, que además obtuvo la mejor precisión (0.31). El segundo mejor  $F_1$  fue 0.25, alcanzado por correferencia genérica y extracción de entidades con LLM (con 3 y 5 runs) y por correferencia *alias-aware* en conjunto con *spaCy* (con 5 runs). El mejor recall fue 0.24 con correferencia genérica en conjunto extracción de entidades con LLM y 3 runs. Véase Tabla 5.4.

Comparado con los mejores resultados globales (con el módulo de correferencia desactivado):  $F_1=0.28$ ,  $recall=0.25$  usando *Shots in System* con lista de entidades y  $runs=3/5/12$  (Tablas 5.1–5.3), este subestudio acorta la brecha pero no la supera. Sí mejora de forma sustantiva el baseline con coreferencia activada de la cuadrícula general ( $F_1$  típico  $\leq 0,12$ ; Tabla 5.2), lo que indica que focalizar la configuración (plantilla y soporte de entidades) eleva el techo cuando el módulo de correferencia está activo.

En conjunto, estos hallazgos sugieren que seguir profundizando la correferencia basada en LLM (con plantillas bien especificadas) aporta más valor que la alternativa con *spaCy-experimental*, aunque el mejor desempeño global del trabajo sigue ubicándose en el escenario sin correferencia.

<sup>1</sup>Ambas variantes son de correferencia por LLM; la genérica no incluye reglas de alias, mientras que la *alias-aware* incorpora reglas operativas de aliasado.

**Tabla 5.4:** Resultados con `gpt-4o-mini`, módulo de correferencia activado, `Shots=System` y enviando la lista de entidades. Se varían corref. genérica vs. alias-aware y extracción de entidades (LLM vs. `spaCy`), con `runs` en {3 y 5}. Mejor valor global en **negrita**, segundo mejor global subrayado.

Corref.	Entidades	Runs	Precisión	Recall	$F_1$
Genérica	spaCy	3	0.27	0.19	0.22
		5	<b>0.31</b>	<u>0.23</u>	<b>0.26</b>
	LLM	3	0.26	<b>0.24</b>	<u>0.25</u>
		5	<u>0.29</u>	<u>0.23</u>	<u>0.25</u>
Alias-aware	spaCy	3	0.25	0.21	0.23
		5	<u>0.29</u>	<u>0.23</u>	<u>0.25</u>
	LLM	3	0.26	0.22	0.24
		5	0.26	0.21	0.23

**Efecto del postprocesamiento de consolidación (deduplicación).** Dado el alto costo de esta etapa (resúmenes por personaje + resumen global implican múltiples llamadas al LLM), se evaluó en un subconjunto de configuraciones previamente prometedoras (todas con *Shots in System* y lista de entidades). La Tabla 5.5 resume los resultados: se reportan Precisión, Recall y  $F_1$  para combinaciones que varían la activación del módulo de correferencia, plantilla de correferencia (genérica vs *alias-aware*) y extracción de entidades (*spaCy* vs *LLM*), con `runs` en {3,5}.

(1) Con correferencia desactivada el postprocesamiento eleva el techo: se alcanza  $F_1=0.30$  con 3 runs ( $+0.02$  absoluto sobre el mejor 0.28 de la cuadrícula global de la Tabla 5.3) y 0.29 con 5 runs. El *recall* sube hasta 0.26 (vs. 0.25 previo), manteniendo una precisión alta (0.34–0.35). (2) Con correferencia activada, el mejor  $F_1$  queda en 0.26, muy por encima del *baseline* de la cuadrícula general con el módulo de correferencia activado (típicamente  $\leq 0.12$ , Tabla 5.2) y en línea con el subestudio controlado (Tabla 5.4). Aquí la plantilla genérica tiende a rendir levemente mejor que *alias-aware* y la extracción de entidades por *spaCy* y por *LLM* quedan parejas en el tope ( $F_1=0.26$ ), con ligeras ventajas de recall del lado de *spaCy* en este set.

La consolidación final basada en LLM sí aporta mejoras medibles, especialmente cuando correferencias están desactivadas, donde se observa el mayor incremento ( $+0.02$ – $+0.03$   $F_1$  y hasta  $+0.01$  en recall). Con el módulo de correferencia activado, la consolidación contribuye a cerrar la brecha respecto del mejor escenario global, aunque no lo supera. Debido a su costo, esta etapa se aplicó sólo a configuraciones fuertes; su integración selectiva parece razonable.

Con el objetivo de contrastar enfoques reportados en la literatura, corrimos dos variantes adicionales sobre el mejor escenario sin correferencia (núcleo operativo): SumAsk y ChatIE. Dado su alto costo en llamadas al LLM, SumAsk se

**Tabla 5.5:** Efecto del postprocesamiento de consolidación (deduplicación de personajes/relaciones). *Shots in System* y con lista de entidades en todos los casos. Mejor Precisión, Recall y  $F_1$  en **negrita**; segundos subrayados.

Coref.	Plantilla coref.	Extr. entidades	Runs	Precisión	Recall	$F_1$
No	Alias-aware	spaCy	3	<b>0.35</b>	<b>0.26</b>	<b>0.30</b>
			5	<u>0.34</u>	<u>0.25</u>	<u>0.29</u>
Sí	Genérica	spaCy	5	0.30	0.24	0.26
		LLM		0.31	0.23	0.26
	Alias-aware	spaCy	5	0.31	0.21	0.25
		LLM		0.26	0.22	0.24

**Tabla 5.6:** Resultados de **SumAsk** y **ChatIE** ejecutados sobre la mejor configuración base (sin correferencia).

Algoritmo	Precisión	Recall	$F_1$
SumAsk	0.03	<b>0.56</b>	0.07
ChatIE	<b>0.26</b>	<b>0.20</b>	<b>0.22</b>

ejecutó una sola vez (sólo para la mejor configuración base), mientras que ChatIE se probó con tres números de corridas. Las columnas *runs*, *shots* y afines no aplican a estos algoritmos, por lo que se omiten del análisis.

SumAsk logra un *recall* muy alto (0.56), pero a costa de una precisión muy baja (0.03), lo que produce un  $F_1$  de 0.07 (Tabla ??). Esto es consistente con su diseño de amplificación por resúmenes y chequeos binarios: recupera muchos verdaderos positivos, pero acumula falsos positivos cuando el contexto es ambiguo o los resúmenes son ruidosos, y su costo en llamadas al LLM es elevado.

ChatIE ofrece un equilibrio más estable: alcanza precisión = 0.26, recall = 0.20 y  $F_1 = 0.22$  (Tabla ??), por debajo del mejor  $F_1 = 0.28$  del flujo base sin correferencia (Tablas 5.1 y 5.3), pero superior al desempeño típico con correferencia activada de la cuadrícula general ( $F_1 \leq 0.12$ ; Tabla 5.2).

En conjunto, ninguno de los dos enfoques supera al mejor escenario base sin correferencia (*Shots in System*, entidades provistas), que mantiene  $F_1 = 0.28$ . SumAsk prioriza cobertura con fuerte costo y caída de precisión; ChatIE resulta un punto medio ( $F_1 = 0.22$ ), útil como línea de base adicional, pero no competitivo frente al mejor flujo.

**Variación de modelos (muestreo acotado por costo)** Con el objetivo de estimar el efecto de modelos más recientes (con mayor capacidad y costo), corrimos una batería acotada de pruebas sobre la combinación más sólida del flujo (correferencia desactivada, *Shots in System*, con lista de entidades), y, en

**Tabla 5.7:** Comparación de modelos con el módulo de correferencia desactivado, *Shots in System* y con lista de entidades. Se reportan precisión, recall y  $F_1$ . Mejor valor global en **negrita**, segundo subrayado.

Modelo	Runs	Precisión	Recall	$F_1$
gpt-4.1-nano	3	0.28	<u>0.20</u>	0.23
	5	0.33	<u>0.18</u>	0.23
gpt-4.1-mini	3	0.38	0.14	0.21
	5	0.42	0.15	0.22
gpt-5-nano	3	<u>0.50</u>	0.08	0.14
	5	<b>0.58</b>	0.07	0.12
gpt-5-mini	3	0.44	<u>0.32</u>	<u>0.37</u>
	5	0.46	<b>0.36</b>	<b>0.40</b>

**Tabla 5.8:** Comparación de modelos con el módulo de correferencia, *Shots in System* y con lista de entidades. Ejecutado con 5 runs. Dos combinaciones fallaron y no se incluyen. Mejor valor global en **negrita**.

Modelo	Precisión	Recall	$F_1$
gpt-4.1-nano	0.30	<b>0.24</b>	<b>0.26</b>
gpt-4.1-mini	<b>0.31</b>	0.23	<b>0.26</b>

menor medida, con correferencia activada para comparar resultados. Dado el costo por inferencia, el muestreo fue limitado.

Los resultados se resumen en las Tablas 5.7 (sin correferencia) y 5.8 (con correferencia). El mejor  $F_1$  global del conjunto se obtuvo con **gpt-5-mini**, módulo de correferencia desactivado, 5 runs:  $F_1 = 0.40$ , acompañado de recall = 0.36 y precisión = 0.46. El segundo mejor  $F_1$  fue 0.37 con **gpt-5-mini**, 3 runs. En contraste, **gpt-5-nano** maximizó precisión (hasta 0.58) pero con *recall* bajo, resultando en  $F_1$  menor.

Con el módulo de correferencia activado y 5 runs, sólo **gpt-4.1-nano** y **gpt-4.1-mini** completaron correctamente, ambos con  $F_1 = 0.26$ , repartiendo el máximo de precisión (**gpt-4.1-mini**, 0.31) y recall (**gpt-4.1-nano**, 0.24). Las variantes **gpt-5-nano/mini** fallaron y no se incluyen.

## 5.5. Síntesis

El mejor desempeño sin postprocesamiento se obtuvo con correferencia desactivada, *Shots in System* y lista de entidades provista, alcanzando  $F_1 = 0.28$  y recall = 0.25 (Tablas 5.1 y 5.3). El segundo mejor resultado fue  $F_1 = 0.26$ ,

también sin correferencia, con *Shots in Chat*. En contraste, al activar la correferencia el rendimiento disminuye notablemente en la cuadrícula general ( $F_1$  típico  $\leq 0,12$ ; Tabla 5.2).

En el subestudio controlado de correferencia por LLM, las mejores configuraciones alcanzaron  $F_1 = 0,26$  (plantilla genérica con spaCy) y precisión = 0.31 / recall = 0.24 (plantilla genérica con extracción de entidades por LLM) (Tabla 5.4). Estos valores reducen la brecha respecto del mejor caso sin correferencia, pero no lo igualan.

Con postprocesamiento de consolidación (deduplicación), el sistema sí supera el mejor caso sin correferencia: se obtiene  $F_1 = 0,30$  y recall = 0.26, con un segundo valor cercano ( $F_1 = 0,29$ ) según Tabla 5.5. La mejora es de hasta +0,03 en  $F_1$ , a costa de mayor uso de LLM.

Los métodos externos muestran patrones distintos: SumAsk maximiza cobertura (recall = 0.56) pero con precisión muy baja ( $F_1 = 0,07$ ), mientras que ChatIE ofrece un balance más estable ( $F_1 = 0,22$ ) (Tabla 5.8). Ninguno supera al mejor flujo *vanilla* sin correferencia.

En conjunto, los resultados indican que *Shots in System* ofrece el mejor equilibrio entre precisión y recall; que aumentar *runs* más allá de 5–8 aporta mejoras marginales; y que la correferencia basada en LLM, aunque superior a alternativas experimentales cuando se evalúa de forma aislada, no desplaza al escenario sin correferencia como mejor opción general. La consolidación postprocesada aparece como un paso selectivo útil cuando se prioriza calidad.

**Nota sobre variación de modelos.** En una evaluación acotada por costos (Sec. 5.4), `gpt-5-mini` elevó el rendimiento del flujo sin correferencia hasta  $F_1 = 0,40$  (recall = 0.36), con un segundo valor cercano ( $F_1 = 0,37$ ). Dado su costo, se reportan como mejoras puntuales, manteniendo `gpt-4o-mini` como referencia principal.

## Capítulo 6

# Conclusiones y trabajo futuro

Este capítulo sintetiza los hallazgos del estudio y pone en perspectiva sus límites y aportes. En particular, resume el rendimiento cuantitativo de las distintas configuraciones evaluadas, contrasta variantes de *prompting* y de orquestación (incluida la correferencia basada en LLM), y discute el impacto del postprocesamiento de consolidación. Además, explicita las principales dificultades técnicas observadas, el valor del corpus anotado y del prototipo como insumos de investigación, y fija una línea de base sobre la cual iterar en trabajos posteriores.

### 6.1. Conclusiones

El objetivo central de este trabajo fue estudiar la viabilidad y límites del uso de modelos de lenguaje (LLM) para la extracción de redes de personajes en textos literarios, con foco específico en la detección de relaciones explícitas entre personajes humanos. En ese marco, se desarrolló un prototipo de aplicación como medio para orquestar el flujo de procesamiento (preprocesamiento, recuperación por *embeddings*, extracción y consolidación) y medio experimental para contrastar configuraciones.

A lo largo de la experimentación se implementaron y compararon mecanismos diversos para la extracción de relaciones con LLM: recuperación semántica con segmentación por tokens, variantes de *prompting* (con y sin lista de entidades, correferencia, pasadas múltiples con deduplicación) y esquemas inspirados en la literatura (*SumAsk*, *ChatIE*). Pese a estas estrategias, reportadas como útiles en otros contextos, los resultados en el dominio narrativo se mantuvieron moderados: el mejor  $F_1$  sin postprocesamiento fue 0.28 (con correferencia desactivada, *Shots in System* y lista de entidades provista por spaCy), mientras que el mejor recall alcanzó 0.25. Por contraste, la mayor precisión (0.39) se obtuvo con *Shots in Chat* y entidades incluidas, pero con  $F_1$  inferior (0,26). Con postprocesamiento de consolidación (deduplicación) aplicado sobre configu-

raciones fuertes, el sistema superó ese techo y alcanzó  $F_1 = 0.30$  y  $\text{recall} = 0.26$  (Tabla 5.5), a costa de un mayor uso de LLM.

Complementariamente, un subestudio con correferencia por LLM (plantillas genérica y *alias-aware*) y comparación entre *spaCy* y extracción de entidades con LLM mostró un máximo de  $F_1 = 0.26$  (genérica y *spaCy*, con 5 runs), con precisión = 0.31, y un máximo de  $\text{recall} = 0.24$  (genérica y extracción de entidades con LLM, 3 runs); estos resultados acortan la brecha respecto del mejor caso sin correferencia, pero no lo superan (Tabla 5.4). En la misma línea, *SumAsk* y *ChatIE* no superaron al flujo base sin correferencia ( $F_1 = 0,28$ ); *SumAsk* privilegia cobertura con fuerte caída de precisión, y *ChatIE* ofrece un equilibrio intermedio ( $F_1 = 0,22$ ). En conjunto, los hallazgos no permiten sostener la viabilidad del enfoque en un escenario *zero-shot* sin apoyos adicionales (conocimiento estructurado, reglas más fuertes o entrenamiento específico). En términos globales, el estudio sugiere que las ventajas de los LLM generalistas no se traducen automáticamente a tareas finas de extracción relacional en narrativa literaria, aun bajo *prompting* cuidadoso y control del contexto.

En línea con estos resultados, es pertinente contextualizar brevemente las principales dificultades técnicas que incidieron en el desempeño (desarrolladas a lo largo del informe): (i) la correferencia presentó obstáculos; (ii) la sensibilidad a instrucciones y contexto propia de los LLM produjo variabilidad según el diseño del *prompt* y la evidencia recuperada; y (iii) las restricciones de ventana de contexto obligaron a segmentar y recuperar selectivamente, exponiendo *trade-offs* entre cobertura y precisión y fenómenos de *lost in the middle*. Estos factores ayudan a explicar el rendimiento observado, aun con mecanismos de consolidación por pasadas múltiples y controles de alias.

Como insumo para una evaluación rigurosa, se construyó un corpus propio anotado a nivel de cuento con seis tipos de relaciones explícitas, incluyendo un registro consistente de *aliases*. Este recurso permitió conducir pruebas controladas, analizar errores y estudiar el impacto de decisiones de diseño (tamaño de fragmento, correferencia) sobre la cobertura y la precisión del sistema.

Los aportes de este trabajo son: (i) un estudio empírico sobre la aplicabilidad de LLM en extracción de redes de personajes en narrativa; (ii) evidencia de que, bajo configuraciones *zero-shot* razonables, los resultados no alcanzan niveles satisfactorios (mejor  $F_1 = 0,28$ ,  $\text{recall} = 0,25$ ) para uso autónomo, si bien la deduplicación eleva el techo a  $F_1 = 0,30$  con mayor costo; y (iii) un corpus anotado y un prototipo que facilitan replicación y extensiones futuras. La interfaz web desarrollada cumple un rol instrumental (inspección, trazabilidad y visualización) y no constituye el resultado principal del estudio.

**Variantes de modelos.** Dado el costo de inferencia, sólo se ejecutó un muestreo reducido con modelos GPT más recientes. Aún con pocas corridas, se observó un techo puntual de  $F_1 = 0.40$  en la configuración más favorable, lo que sugiere que modelos de mayor capacidad pueden elevar el rendimiento bajo el mismo pipeline y datos. Sin embargo, por restricciones de presupuesto y para mantener comparabilidad, el análisis principal se centró en `gpt-4o-mini` y

configuraciones de operativa sostenida.

Por último, si bien la inclusión de LLM generalistas en configuración *zero-shot* puso de manifiesto sus límites en extracción relacional, esto no alcanza para descartar su utilidad potencial. La evolución reciente de estos modelos y el surgimiento de técnicas que simplifican la tarea a resolver (descomposición en pasos, alineamiento con tareas establecidas, postprocesamiento y refinamiento de resultados, y reglas de alias coherentes) sugieren que la combinación de mejores prácticas identificadas durante la exploración puede fortalecer el sistema, ya sea mediante apoyos simbólicos y validación estructurada o con ajustes específicos al dominio narrativo. Asimismo, la rápida evolución de los LLM y del ecosistema de herramientas habilita la aparición de métodos más adecuados o la mejora de la misma arquitectura aquí propuesta mediante modelos más avanzados (ventanas de contexto mayores, mejor razonamiento estructurado, salidas con restricciones/validadores y recuperación más focalizada). En consecuencia, los resultados deben leerse como una línea de base sobre la cual iterar: la incorporación progresiva de nuevas capacidades y librerías especializadas podría traducirse en mejoras sustantivas de cobertura y precisión sin resignar trazabilidad ni control de costos.

## 6.2. Trabajo Futuro

A partir del desarrollo y la evaluación del sistema, se identifican varias líneas de avance que pueden mejorar tanto el rendimiento como la robustez de la solución.

En primer lugar, y dada la naturaleza del dominio narrativo, se considera prioritario incorporar mecanismos para identificar y modelar la evolución temporal de las relaciones a lo largo del texto. En muchas obras, los vínculos entre personajes no son estáticos: cambian, se transforman o incluso se invierten con el progreso de la trama. Integrar esta dimensión implica segmentar el contenido por unidades narrativas (capítulos, escenas o ventanas deslizantes), detectar eventos relevantes y representar la evolución en grafos dinámicos que distingan entre relaciones iniciales, transitorias y finales.

En paralelo, se propone ampliar y diversificar el corpus de evaluación, incorporando textos de géneros, estilos y estructuras argumentales diferentes. Una base empírica más rica permitiría contrastar la capacidad de generalización y validar la robustez del sistema frente a desafíos lingüísticos variados (registro, arcaísmos, diálogos extensos, subtramas), además de habilitar comparaciones entre obras, autores o períodos y el procesamiento de colecciones completas.

En cuanto a la arquitectura de modelado, la rápida evolución de los LLM sugiere evaluar modelos más recientes con ventanas de contexto ampliadas y mayor capacidad de razonamiento estructurado. Existen actualmente modelos comerciales y abiertos con mayor contexto y/o rendimiento, pero no fueron utilizados en este proyecto por restricciones de presupuesto (mayor costo por inferencia) y de infraestructura (las alternativas que podrían ejecutarse localmente demandan recursos de cómputo no disponibles). Su adopción podría reducir la necesidad de

segmentación agresiva y mitigar fenómenos de *lost in the middle*, favoreciendo una comprensión más coherente de pasajes extensos.

De forma complementaria, sería pertinente comparar sistemáticamente modelos encoder–decoder (por ejemplo, T5 o BART), ampliamente utilizados en extracción de información por su idoneidad para transformar texto en salidas estructuradas. Aun con las dificultades que presentan para textos largos, podrían explorarse esquemas por ventanas con memoria global o resúmenes intermedios.

También es conveniente seguir explorando la correferencia basada en LLM y su interacción con la extracción de entidades con LLM, aunque no se considera la línea más prometedora en el corto plazo. En el subestudio controlado, las plantillas genéricas mostraron un rendimiento competitivo frente a variantes *alias-aware* y mejoraron el piso observado con correferencia basada sólo en *spaCy-experimental*, pero sin superar el desempeño global de los escenarios sin correferencia.

Dado que el postprocesamiento de consolidación (deduplicación/normalización) elevó el techo de  $F_1$  hasta 0.30 en escenarios fuertes, conviene profundizar técnicas de consolidación eficientes: estrategias de *caching* y reutilización de resúmenes, heurísticas de fusión, *early-exit* por confianza y validadores estructurados, de modo de mantener el beneficio sin disparar el costo en tokens/llamadas.

Finalmente, el ecosistema de herramientas y modelos avanza con rapidez, por lo que es razonable anticipar utilidades más adecuadas para memoria de largo alcance, validación estructural o indexación semántica. La arquitectura propuesta está preparada para integrar estos componentes de forma incremental, preservando trazabilidad y control de costos. Como complemento metodológico, futuras evaluaciones deberían incluir análisis de error sistemático, tests de significación estadística y curvas costo–rendimiento (tokens/llamadas vs.  $F_1$ ) para guiar decisiones bajo presupuesto.

En conjunto, una base empírica más diversa, modelos con mayor capacidad y orquestación más precisa.

## Siglas

- **ACE05** — *Automatic Content Extraction 2005*
- **ACID** — Atomicidad, Consistencia, Aislamiento, Durabilidad (*Atomicity, Consistency, Isolation, Durability*)
- **ADE** — *Adverse Drug Event* (corpus biomédico)
- **API** — Interfaz de Programación de Aplicaciones (*Application Programming Interface*)
- **BART** — *Bidirectional and Auto-Regressive Transformers* (*Transformadores Bidireccionales y Autorregresivos*)
- **BERT** — *Bidirectional Encoder Representations from Transformers* (*Representaciones Bidireccionales Codificadas por Transformers*)
- **CoNLL** — *Conference on Natural Language Learning* (p.ej., CoNLL 2004/2012)
- **CoT** — Cadena de Razonamiento (*Chain of Thought*)
- **CPU** — Unidad Central de Procesamiento (*Central Processing Unit*)
- **CSRF** — Falsificación de Solicitud en Sitios Cruzados (*Cross-Site Request Forgery*)
- **CSV** — Valores Separados por Comas (*Comma-Separated Values*)
- **DocRED** — *Document-Level Relation Extraction Dataset*
- $F_1$  — Medida  $F_1$  (media armónica de Precisión y Recall) ( *$F_1$  score*)
- **GPT** — *Generative Pre-trained Transformer* (*Transformador Generativo Preentrenado*)
- **GPU** — Unidad de Procesamiento Gráfico (*Graphics Processing Unit*)
- **IE** — Extracción de Información (*Information Extraction*)
- **JSON** — Notación de Objetos de JavaScript (*JavaScript Object Notation*)
- **k-NN** — k Vecinos Más Cercanos (*k-Nearest Neighbors*)
- **LLM(s)** — Modelo(s) de Lenguaje de Gran Tamaño (*Large Language Model(s)*)
- **LLaMA** — *Large Language Model Meta AI* (*Modelo de Lenguaje de Gran Tamaño de Meta AI*)
- **MVC** — Modelo–Vista–Controlador (*Model–View–Controller*)

- **NER** — Reconocimiento de Entidades con Nombre (*Named Entity Recognition*)
- **NYT** — *New York Times* (corpus anotado de relaciones)
- **ORM** — Mapeo Objeto-Relacional (*Object-Relational Mapping*)
- **P** — Precisión (*Precision*)
- **PLN** — Procesamiento del Lenguaje Natural (*Natural Language Processing, NLP*)
- **POS** — Categorías gramaticales / Etiquetado morfosintáctico (*Part-of-Speech*)
- **QA** — Preguntas y Respuestas (*Question Answering*)
- **R** — Cobertura / Exhaustividad (*Recall*)
- **RAG** — Generación Aumentada por Recuperación (*Retrieval-Augmented Generation*)
- **RE** — Extracción de Relaciones (*Relation Extraction*)
- **Re-DocRED** — *Revised DocRED*
- **Re-TACRED** — *Revised TACRED*
- **RoBERTa** — *Robustly Optimized BERT Approach (Enfoque BERT Optimiz. Robustamente)*
- **SciERC** — *Scientific Information Extraction from Research Papers*
- **SemEval** — *Semantic Evaluation* (tareas compartidas de evaluación semántica)
- **SQL** — Lenguaje de Consulta Estructurado (*Structured Query Language*)
- **SPA** — Aplicación de Página Única (*Single Page Application*)
- **T5** — *Text-to-Text Transfer Transformer (Transformador de Transferencia Texto-a-Texto)*
- **TACRED** — *TAC Relation Extraction Dataset*
- **TACREV** — *TACRED Revisited*
- **UI** — Interfaz de Usuario (*User Interface*)
- **USE** — Codificador Universal de Oraciones (*Universal Sentence Encoder*)
- **UX** — Experiencia de Usuario (*User eXperience*)
- **XSS** — *Scripting en Sitios Cruzados (Cross-Site Scripting)*

## Glosario

- **Alias** (*alias*): Variante superficial o denominación alternativa de una misma entidad (p. ej., “Mr. Holmes”/“Holmes”).
- **Chain of Thought (CoT)**: Técnica de prompting que explicita pasos de razonamiento intermedios para mejorar consistencia.
- **Consolidación** (*consolidation*): Paso de fusión y depuración posterior a la extracción (unificación de entidades y deduplicación de relaciones).
- **Correferencia** (*coreference resolution*): Identificación de menciones distintas que refieren a la misma entidad.
- **Deduplicación** (*deduplication*): Eliminación de entradas redundantes (entidades/relaciones duplicadas).
- **Distancia coseno** (*cosine distance*): Medida de disimilitud basada en el ángulo entre dos vectores normalizados.
- **Distancia coseno vs. distancia euclidiana** (*cosine similarity vs. euclidean distance*): La *similitud coseno* mide el ángulo entre vectores (robusta a escalas), mientras que la *distancia euclidiana* mide diferencia absoluta de magnitudes. *Ejemplo*: si  $\vec{a} = \lambda \vec{b}$  con  $\lambda > 0$ , entonces  $\cos(\vec{a}, \vec{b}) = 1$  (máxima similitud), pero la distancia euclidiana  $\|\vec{a} - \vec{b}\|$  crece con  $\lambda$ . En *embeddings* normalizados, la coseno captura mejor la afinidad semántica independiente de la norma.
- **Embedding** (*vector embedding*): Representación numérica densa de un texto en un espacio vectorial donde la cercanía refleja similitud semántica.
- $F_1$  (*F1 score*): Media armónica entre precisión y recall; resume el equilibrio entre ambas.
- **Few-shot** (*few-shot learning*): Configuración donde se proporcionan pocos ejemplos dentro de la interacción para ilustrar la tarea.
- **Fine-tuning** (*fine-tuning*): Ajuste adicional de un modelo preentrenado en un conjunto específico para una tarea concreta.
- **Grafo** (*graph*): Estructura formada por nodos (entidades) y aristas (relaciones) que representan redes de interacción.
- **Grafo dinámico** (*dynamic graph*): Grafo cuya estructura o pesos cambian a lo largo del tiempo o de la narrativa.
- **Grafo tipado** (*typed graph*): Grafo cuyas aristas y/o nodos poseen tipos o etiquetas pertenecientes a un conjunto finito.
- **Instruction-tuning** (*instruction tuning*): Ajuste de LLM en grandes conjuntos de datos de seguimiento de instrucciones.

- **k-NN** (*k-nearest neighbors*): Selección de los  $k$  elementos más cercanos a un vector consulta en el espacio de embeddings.
- **Lost in the middle**: Fenómeno por el cual los modelos atienden menos a la información ubicada en posiciones intermedias del contexto.
- **Modelo *decoder-only*** (*decoder-only transformer*): Arquitectura autoregresiva típica en LLM generativos.
- **Modelo *encoder-decoder*** (*sequence-to-sequence transformer*): Arquitectura con codificador y decodificador, usada en T5/BART.
- **NER** (*Named Entity Recognition*): Tarea de detección y rotulado de entidades con nombre en texto.
- **Normalización** (*normalization*): Conjunto de reglas para homogeneizar texto o entidades.
- **Postprocesamiento** (*post-processing*): Transformaciones aplicadas luego de la inferencia para mejorar calidad o coherencia.
- **Precisión** (*precision*): Proporción de aciertos sobre las predicciones positivas del sistema.
- **Prompt** (*prompt*): Instrucciones y/o ejemplos entregados al modelo.
- **Prompting** (*prompting*): Diseño sistemático de *prompts*.
- **RAG** (*Retrieval-Augmented Generation*): Paradigma donde se incorporan fragmentos recuperados.
- **Recall** (*recall*): Proporción de aciertos sobre las instancias positivas reales.
- **Recuperación** (*retrieval*): Selección de fragmentos de texto relevantes a partir de una consulta.
- **Resumen narrativo** (*narrative summary*): Síntesis global del argumento, personajes y contexto.
- **Resumen por personaje** (*character summary*): Síntesis focalizada en rasgos o rol de un personaje.
- **Segmento** (*chunk/segment*): Fragmento de texto utilizado para inferencia.
- **Self-consistency** (*auto-consistency*): Estrategia de ejecutar varias veces y consolidar resultados.
- **spaCy** (*spaCy*): Biblioteca de PLN en Python para NER, tokenización, dependencias, etc.

- **supporting\_text** (ES/EN): Campo obligatorio en la salida de extracción de relaciones que contiene texto de evidencia.
- **Text-to-image** (*text-to-image*): Generación de imágenes a partir de texto.
- **Tiktoken** (*tiktoken*): Biblioteca de tokenización compatible con modelos OpenAI.
- **Token** (*token*): Unidad mínima con la que opera un modelo.
- **Tokenización** (*tokenization*): Proceso de dividir texto en tokens.
- **Ventana de contexto** (*context window*): Longitud máxima de tokens procesables en una entrada.
- **Ventana de tokens efectiva** (*effective context window*): Porción utilizable real de la ventana de contexto.
- **Ventana deslizante** (*sliding window*): Estrategia de segmentación con solapamiento.
- **Zero-shot** (*zero-shot learning*): Configuración sin ejemplos proporcionados.



# Referencias

- Agarwal, A., Kotalwar, A., y Rambow, O. (2013, October). Automatic extraction of social networks from literary text: A case study on alice in wonderland. En *Proceedings of the sixth international joint conference on natural language processing (ijcnlp 2013)* (pp. 1202–1208). Nagoya, Japan: Asian Federation of Natural Language Processing. Descargado de <https://aclanthology.org/I13-1171/>
- Alberich, R., Miró-Julià, J., y Rosselló, F. (2002). *Marvel universe looks almost like a real social network*. arXiv preprint cond-mat/0202174. Descargado de <https://arxiv.org/abs/cond-mat/0202174>
- Allen Institute for Artificial Intelligence (AI2). (2022). *allenai/allennlp: An open-source nlp research library built on pytorch*. GitHub repository. (Disponible en: <https://github.com/allenai/allennlp>, último acceso: 1 de septiembre de 2025)
- Allen Institute for Artificial Intelligence (AI2). (2025). *Allennlp documentation (v2.10.1)*. Documentation website. (Disponible en: <https://docs.allennlp.org/main/>, último acceso: 1 de septiembre de 2025)
- Alt, C., Gabryszak, A., y Hennig, L. (2020, julio). TACRED revisited: A thorough evaluation of the TACRED relation extraction task. En D. Jurafsky, J. Chai, N. Schlueter, y J. Tetreault (Eds.), *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 1558–1569). Online: Association for Computational Linguistics. Descargado de <https://aclanthology.org/2020.acl-main.142/> doi: 10.18653/v1/2020.acl-main.142
- Amalvy, A., Labatut, V., y Dufour, R. (2024a). Renard: A modular pipeline for extracting character networks from narrative texts. *Journal of Open Source Software*, 9(98), 6574. Descargado de <https://doi.org/10.21105/joss.06574> (Under review at the Journal of Open Source Software (JOSS)) doi: 10.21105/joss.06574
- Amalvy, A., Labatut, V., y Dufour, R. (2024b). *The role of natural language processing tasks in automatic literary character network construction*.

- arXiv preprint arXiv:2412.11560. Descargado de <https://arxiv.org/abs/2412.11560>
- Artificial Analysis. (2025). *Flux – quality, generation time & price analysis*. Artificial Analysis website. (Disponible en: <https://artificialanalysis.ai/text-to-image/model-family/flux>, último acceso: 1 de septiembre de 2025)
- Bazzan, A. L. C. (2020). I will be there for you: Clique, character centrality, and community detection in friends. *Computational and Applied Mathematics*, 39(3), 192. Descargado de <https://doi.org/10.1007/s40314-020-01222-7> doi: 10.1007/s40314-020-01222-7
- Bearman, P. S., y Stovel, K. (2000). Becoming a nazi: A model for narrative networks. *Poetics*, 27(2), 69–90. Descargado de [https://doi.org/10.1016/S0304-422X\(99\)00022-4](https://doi.org/10.1016/S0304-422X(99)00022-4) doi: 10.1016/S0304-422X(99)00022-4
- Bossaert, G., y Meidert, N. (2013). “we are only as strong as we are united, as weak as we are divided”: A dynamic analysis of the peer support networks in the harry potter books. *Open Journal of Applied Sciences*, 3(2), 174–185. Descargado de <https://www.scirp.org/journal/paperinformation?paperid=33365> doi: 10.4236/ojapps.2013.32024
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... others (2020). Language models are few-shot learners. En *Advances in neural information processing systems (neurips)* (Vol. 33, pp. 1877–1901). Curran Associates, Inc. Descargado de <https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf>
- Canário, T. G., Duarte, C., Pinheiro, F. L., y Pereira, J. L. M. (2025). *Taggus: An automated pipeline for the extraction of characters’ social networks from portuguese fiction literature*. arXiv preprint arXiv:2508.03358. Descargado de <https://arxiv.org/abs/2508.03358>
- Choi, Y.-M., y Kim, H.-J. (2007). A directed network of greek and roman mythology. *Physica A: Statistical Mechanics and its Applications*, 382(2), 665–671. Descargado de <https://www.sciencedirect.com/science/article/pii/S0378437107003901> doi: 10.1016/j.physa.2007.04.035
- Chun, Y. E., Hwang, T., won Hwang, S., y Kim, B.-H. (2025). *Crefit: Sequential multi-agent llm for character relation extraction*. Descargado de <https://arxiv.org/abs/2505.24553> (arXiv:2505.24553)
- comfyanonymous. (2023). *Comfyui: A modular gui/api for building diffusion model workflows*. GitHub repository. (Disponible en: <https://github.com/comfyanonymous/ComfyUI>, último acceso: 1 de septiembre de 2025)
- DeBuse, M. A., y Warnick, S. (2023). Plot extraction and the visualization of narrative flow. *Natural Language Engineering*. doi: 10.1017/S1351324923000232

- DeBuse, M. A., y Warnick, S. (2024). Plot extraction and the visualization of narrative flow. *Natural Language Engineering*, 30(3), 480–524. Descargado de <https://doi.org/10.1017/S1351324923000232> doi: 10.1017/S1351324923000232
- Detroja, K., Bhensdadia, C., y Bhatt, B. (2022). A survey on relation extraction. *SSRN Electronic Journal*. Descargado de <https://doi.org/10.2139/ssrn.4173454> doi: 10.2139/ssrn.4173454
- Devlin, J., Chang, M.-W., Lee, K., y Toutanova, K. (2019). *Bert: Pre-training of deep bidirectional transformers for language understanding*. Proceedings of NAACL-HLT 2019. (Disponible: <https://research.google/pubs/bert-pre-training-of-deep-bidirectional-transformers-for-language-understanding/>, último acceso: 1 de septiembre de 2025)
- Diaz-Garcia, J. A., y Lopez, J. A. D. (2025). A survey on cutting-edge relation extraction techniques based on language models. *Artificial Intelligence Review*, 58(9), 287.
- Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., y Weischedel, R. (2004, May). The automatic content extraction (ace) program: Tasks, data, and evaluation. En *Proceedings of the fourth international conference on language resources and evaluation (lrec'04)*. Lisbon, Portugal: European Language Resources Association (ELRA). Descargado de <https://aclanthology.org/L04-1011/>
- Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., ... others (2022). *A survey on in-context learning*. arXiv preprint arXiv:2301.00234. Descargado de <https://arxiv.org/abs/2301.00234>
- Doyle, A. C. (2025). *Works by arthur conan doyle (digital edition)*. Project Gutenberg. (Disponible en: <https://www.gutenberg.org/ebooks/author/69>, edición digital por Project Gutenberg, último acceso: 1 de septiembre de 2025)
- Explosion AI. (2023). *spacy-experimental: Cutting-edge experimental spacy components and features*. GitHub repository. (Disponible en: <https://github.com/explosion/spacy-experimental>, último acceso: 1 de septiembre de 2025)
- Explosion AI. (2025). *Using spacy with large language models*. <https://spacy.io/usage/large-language-models>. (Último acceso: 1 de septiembre de 2025)
- Gurulingappa, H., Rajput, A. M., Roberts, A., Fluck, J., Hofmann-Apitius, M., y Toldo, L. (2012). Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics*, 45(5), 885-892. Descargado de <https://www.sciencedirect.com/science/article/pii/S1532046412000615> (Text

Mining and Natural Language Processing in Pharmacogenomics) doi: <https://doi.org/10.1016/j.jbi.2012.04.008>

- Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., . . . Szpakowicz, S. (2010, julio). SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. En K. Erk y C. Strapparava (Eds.), *Proceedings of the 5th international workshop on semantic evaluation* (pp. 33–38). Uppsala, Sweden: Association for Computational Linguistics. Descargado de <https://aclanthology.org/S10-1006/>
- Hettinger, L., Becker, M., Reger, I., Jannidis, F., y Hotho, A. (2015). Genre classification on german novels. En *Proceedings of the 26th international workshop on database and expert systems applications (dexa 2015)* (pp. 249–253). Descargado de <https://doi.org/10.1109/DEXA.2015.62> doi: 10.1109/DEXA.2015.62
- Hugging Face / spaCy Universe. (2025). *Neuralcoref – state-of-the-art coreference resolution based on neural nets and spacy*. spaCy Universe project page. (Disponible en: <https://spacy.io/universe/project/neuralcoref>, último acceso: 1 de septiembre de 2025)
- Jeong, S. W., y Ročková, V. (2025). *From small to large language models: Revisiting the federalist papers*. arXiv preprint arXiv:2503.01869. Descargado de <https://arxiv.org/abs/2503.01869>
- Jockers, M. L. (2013). *Macroanalysis: Digital methods and literary history*. University of Illinois Press. Descargado de <https://doi.org/10.5406/illinois/9780252037528.001.0001> doi: 10.5406/illinois/9780252037528.001.0001
- Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., y Levy, O. (2020). Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8, 64–77.
- Jurafsky, D., y Martin, J. H. (2008). *Speech and language processing: An introduction to speech recognition, computational linguistics and natural language processing*. Upper Saddle River, NJ: Prentice Hall.
- Jurafsky, D., y Martin, J. H. (2025). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition* (3rd ed.). Self-published online manuscript. Descargado de <https://web.stanford.edu/~jurafsky/slp3/> (Online manuscript, released August 24, 2025)
- Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., . . . Amodi, D. (2020). *Scaling laws for neural language models*. arXiv preprint arXiv:2001.08361. Descargado de <https://arxiv.org/abs/2001.08361>

- Knuth, D. E. (1994). *The stanford graphbase: A platform for combinatorial computing*. ACM Press / Addison-Wesley. Descargado de <https://www-cs-faculty.stanford.edu/~knuth/sgb.html>
- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., y Iwasawa, Y. (2022). Large language models are zero-shot reasoners. En *Advances in neural information processing systems (neurips 2022)* (Vol. 35, pp. 22199–22213). Curran Associates, Inc. Descargado de <https://arxiv.org/pdf/2205.11916.pdf>
- Labatut, V., y Bost, X. (2019). Extraction and analysis of fictional character networks: A survey. *ACM Computing Surveys*, 52(5), 89. doi: 10.1145/3344548
- Li, G., Wang, P., y Ke, W. (2023). *Revisiting large language models as zero-shot relation extractors*. arXiv preprint arXiv:2310.05028. Descargado de <https://arxiv.org/abs/2310.05028>
- Li, J., Jia, Z., y Zheng, Z. (2023). Semi-automatic data enhancement for document-level relation extraction with distant supervision from large language models. En *Proceedings of the 2023 conference on empirical methods in natural language processing* (pp. 5495–5505). Association for Computational Linguistics. Descargado de <https://aclanthology.org/2023.emnlp-main.334/>
- Li, X., Chen, K., Long, Y., y Zhang, M. (2024). *Llm with relation classifier for document-level relation extraction*. arXiv preprint arXiv:2408.13889. Descargado de <https://arxiv.org/abs/2408.13889>
- 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*. arXiv preprint arXiv:2307.03172. Descargado de <https://arxiv.org/abs/2307.03172>
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). *Roberta: A robustly optimized bert pretraining approach*. arXiv preprint arXiv:1907.11692. Descargado de <https://arxiv.org/abs/1907.11692>
- Luan, Y., He, L., Ostendorf, M., y Hajishirzi, H. (2018, octubre-noviembre). Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. En E. Riloff, D. Chiang, J. Hockenmaier, y J. Tsujii (Eds.), *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 3219–3232). Brussels, Belgium: Association for Computational Linguistics. Descargado de <https://aclanthology.org/D18-1360/> doi: 10.18653/v1/D18-1360
- Martinelli, G., Barba, E., y Navigli, R. (2024). *Maverick: Efficient and accurate coreference resolution defying recent trends*. arXiv preprint arXiv:2407.21489. Descargado de <https://arxiv.org/abs/2407.21489>

- Moretti, F. (2011). *Network theory, plot analysis* (Inf. Téc. n.º Pamphlet 2). Stanford Literary Lab. Descargado de <https://litlab.stanford.edu/LiteraryLabPamphlet2.pdf>
- OpenAI. (2025a). *Gpt models: Api reference and documentation*. OpenAI API documentation. (Disponible en: <https://platform.openai.com/docs/models>, último acceso: 1 de septiembre de 2025)
- OpenAI. (2025b). *tiktoken: a fast bpe tokenizer for use with openai's models*. GitHub repository. (Disponible en: <https://github.com/openai/tiktoken>, último acceso: 1 de septiembre de 2025)
- Park, S.-B., Oh, K.-J., y Jo, G.-S. (2012, jul). Social network analysis in a movie using character-net. *Multimedia Tools and Applications*, 59(2), 601–627. Descargado de <https://doi.org/10.1007/s11042-011-0725-1> doi: 10.1007/s11042-011-0725-1
- Prado, S. D., Dahmen, S. R., Bazzan, A. L. C., Mac Carron, P., y Kenna, R. (2016). *Temporal network analysis of literary texts*. arXiv preprint arXiv:1602.07275. Descargado de <https://arxiv.org/abs/1602.07275>
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... Sutskever, I. (2021). Learning transferable visual models from natural language supervision. En *Proceedings of the 38th international conference on machine learning (icml 2021)* (pp. 8748–8763). PMLR. Descargado de <https://proceedings.mlr.press/v139/radford21a.html>
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140), 1–67. Descargado de <https://www.jmlr.org/papers/v21/20-074.html>
- Ramos-Flores, O., y Pinto, D. (2021). Identificación y extracción de relaciones entre entidades empleando árboles de dependencia. *Revista Colombiana de Computación*, 22, 22–36. Descargado de <https://doi.org/10.29375/25392115.4294> doi: 10.29375/25392115.4294
- Riedel, S., Yao, L., y McCallum, A. (2010). Modeling relations and their mentions without labeled text. En *Machine learning and knowledge discovery in databases (ecml pkdd 2010)* (Vol. 6323, pp. 148–163). Berlin, Heidelberg: Springer. Descargado de [https://link.springer.com/chapter/10.1007/978-3-642-15939-8\\_10](https://link.springer.com/chapter/10.1007/978-3-642-15939-8_10) doi: 10.1007/978-3-642-15939-8\_10
- Roth, D., y Yih, W.-t. (2004). A linear programming formulation for global inference in natural language tasks. En *Proceedings of the eighth conference on computational natural language learning (CoNLL-2004) at HLT-NAACL 2004* (pp. 1–8). Boston, Massachusetts, USA: Association for Computational Linguistics. Descargado de <https://aclanthology.org/W04-2401/>

- Sarkar, A., Drosos, I., Deline, R., Gordon, A. D., Negreanu, C., Rintel, S., ... Zorn, B. (2023). *Participatory prompting: A user-centric research method for eliciting ai assistance opportunities in knowledge workflows*. arXiv preprint arXiv:2312.16633. Descargado de <https://arxiv.org/abs/2312.16633>
- Schulhoff, S., Ilie, M., Balepur, N., Kahadze, K., Liu, A., Si, C., ... Resnik, P. (2025). *The prompt report: A systematic survey of prompt engineering techniques*. arXiv preprint arXiv:2406.06608. Descargado de <https://arxiv.org/abs/2406.06608>
- Seow, W. L., Chaturvedi, I., Hogarth, A., Mao, R., y Cambria, E. (2025). A review of named entity recognition: From learning methods to modelling paradigms and tasks. *Artificial Intelligence Review*, 58(10), 1–87. doi: 10.1007/s10462-025-10802-6
- Smeets, R. (2021). *Character constellations: Representations of social groups in present-day dutch literary fiction*. Leuven: Leuven University Press. Descargado de <http://www.jstor.org/stable/j.ctv21wj5cb>
- Speer, R. (2019). *ftfy*. Zenodo. Descargado de <https://doi.org/10.5281/zenodo.2591652> (Version 5.5) doi: 10.5281/zenodo.2591652
- Stiller, J., Nettle, D., y Dunbar, R. I. M. (2003). The small world of shakespeare's plays. *Human Nature*, 14(4), 397–408. Descargado de <https://doi.org/10.1007/s12110-003-1013-1> doi: 10.1007/s12110-003-1013-1
- Stoica, G., Platanios, E. A., y Poczos, B. (2021, May). Re-tacred: Addressing shortcomings of the tacred dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15), 13843-13850. Descargado de <https://ojs.aaai.org/index.php/AAAI/article/view/17631> doi: 10.1609/aaai.v35i15.17631
- Sudhahar, S., De Fazio, G., Franzosi, R., y Cristianini, N. (2015). Network analysis of narrative content in large corpora. *Natural Language Engineering*, 21(1), 81–112. Descargado de <https://doi.org/10.1017/S1351324913000247> doi: 10.1017/S1351324913000247
- Suen, C., Kuenzel, L., y Castelltort Gil, S. (2013). Extraction and analysis of character interaction networks from plays and movies. En *Proceedings of the digital humanities conference (dh 2013)*. Lincoln, Nebraska, USA. Descargado de <https://api.semanticscholar.org/CorpusID:9383329>
- Swarup, A., Pan, T., Wilson, R., Bhandarkar, A., y Woodard, D. (2025, January). LLM4RE: A data-centric feasibility study for relation extraction. En *Proceedings of the 31st international conference on computational linguistics (coling 2025)* (pp. 6670–6691). Abu Dhabi, UAE: Association for Computational Linguistics. Descargado de <https://aclanthology.org/2025.coling-main.447/>

- Tan, Q., Xu, L., Bing, L., Ng, H. T., y Aljunied, S. M. (2022, diciembre). Revisiting DocRED - addressing the false negative problem in relation extraction. En Y. Goldberg, Z. Kozareva, y Y. Zhang (Eds.), *Proceedings of the 2022 conference on empirical methods in natural language processing* (pp. 8472–8487). Abu Dhabi, United Arab Emirates: Association for Computational Linguistics. Descargado de <https://aclanthology.org/2022.emnlp-main.580/> doi: 10.18653/v1/2022.emnlp-main.580
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, B., ... others (2023). *Llama 2: Open foundation and fine-tuned chat models*. arXiv preprint arXiv:2307.09288. Descargado de <https://arxiv.org/abs/2307.09288>
- Trovati, M., y Brady, J. (2014). Towards an automated approach to extract and compare fictional networks: An initial evaluation. En *Proceedings of the 25th international workshop on database and expert systems applications (dexa 2014)* (pp. 246–250). IEEE. Descargado de <https://doi.org/10.1109/DEXA.2014.58> doi: 10.1109/DEXA.2014.58
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. En *Advances in neural information processing systems* (Vol. 30). Descargado de <https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>
- Voloshinov, A. V., y Gozhanskaya, I. V. (2009). Russian vs. english drama in the context of network theory. En *Proceedings of xxth international association of empirical aesthetics biennial congress* (pp. 19–22). Descargado de <https://api.semanticscholar.org/CorpusID:209753936>
- Walker, C., Strassel, S., Medero, J., y Maeda, K. (2006). *Ace 2005 multilingual training corpus (ldc2006t06)*. Web Download. Philadelphia, PA. Descargado de <https://catalog.ldc.upenn.edu/LDC2006T06> doi: 10.35111/mwxc-vh88
- Wan, Z., Cheng, F., Mao, Z., Liu, Q., Song, H., Li, J., y Kurohashi, S. (2023). *Gpt-re: In-context learning for relation extraction using large language models*. arXiv preprint arXiv:2305.02105. Descargado de <https://arxiv.org/abs/2305.02105>
- Wang, L., Liu, D., Liu, X., y He, X. (2025). Scaling down text encoders of text-to-image diffusion models. En *Proceedings of the cvpr 2025* (pp. 18424–18433). Descargado de [https://openaccess.thecvf.com/content/CVPR2025/papers/Wang\\_Scaling\\_Down\\_Text\\_Encoders\\_of\\_Text-to-Image\\_Diffusion\\_Models\\_CVPR\\_2025\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2025/papers/Wang_Scaling_Down_Text_Encoders_of_Text-to-Image_Diffusion_Models_CVPR_2025_paper.pdf) doi: 10.1109/CVPRW56359.2025.02030
- Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., ... Zhou, D. (2022). *Self-consistency improves chain of thought reasoning in language models*. arXiv preprint arXiv:2203.11171. Descargado de <https://arxiv.org/abs/2203.11171>

- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., ... others (2022). Chain-of-thought prompting elicits reasoning in large language models. En *Advances in neural information processing systems (neurips 2022)* (Vol. 35, pp. 24824–24837). Curran Associates, Inc. Descargado de <https://arxiv.org/pdf/2201.11903.pdf>
- Wei, X., Cui, X., Cheng, N., Wang, X., Zhang, X., Huang, S., ... others (2023). *Chatie: Zero-shot information extraction via chatting with chatgpt*. arXiv preprint arXiv:2302.10205. Descargado de <https://arxiv.org/abs/2302.10205>
- Weng, C.-Y., Chu, W.-T., y Wu, J.-L. (2007). Movie analysis based on roles' social network. En *2007 IEEE International Conference on Multimedia and Expo* (pp. 1403–1406). doi: 10.1109/ICME.2007.4284922
- Wiggins, W. F., y Tejani, A. S. (2022). On the opportunities and risks of foundation models for natural language processing in radiology. *Radiology: Artificial Intelligence*, 4(4), e220119.
- Xu, D., Chen, W., Peng, W., Zhang, C., Xu, T., Zhao, X., ... Chen, E. (2024). Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18(6), 186357.
- Xue, L., Zhang, D., Dong, Y., y Tang, J. (2024). *Autore: Document-level relation extraction with large language models*. arXiv preprint arXiv:2403.14888. Descargado de <https://arxiv.org/abs/2403.14888>
- Yan, Y., Zhao, H., Zhu, S., Liu, H., Zhang, Z., y Jia, Y. (2025). *Dialogue-based multi-dimensional relationship extraction from novels*. arXiv preprint arXiv:2507.04852. Descargado de <https://arxiv.org/abs/2507.04852>
- Yao, Y., Ye, D., Li, P., Han, X., Lin, Y., Liu, Z., ... Sun, M. (2019). Docred: A large-scale document-level relation extraction dataset. En *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 764–777). Florence, Italy: Association for Computational Linguistics. Descargado de <https://aclanthology.org/P19-1074/> doi: 10.18653/v1/P19-1074
- Zhang, C., Peng, B., Sun, X., Niu, Q., Liu, J., Chen, K., ... Wang, T. (2024). *From word vectors to multimodal embeddings: Techniques, applications, and future directions for large language models*. arXiv preprint arXiv:2411.05036. Descargado de <https://arxiv.org/abs/2411.05036>
- Zhang, K., Gutiérrez, B. J., y Su, Y. (2023). *Aligning instruction tasks unlocks large language models as zero-shot relation extractors*. arXiv preprint arXiv:2305.11159. Descargado de <https://arxiv.org/abs/2305.11159>
- Zhang, N., y Tang, H. (2024). *Text-to-image synthesis: A decade survey*. arXiv preprint arXiv:2411.16164. Descargado de <https://arxiv.org/abs/2411.16164>

- Zhang, Y., Zhong, V., Chen, D., Angeli, G., y Manning, C. D. (2017, septiembre). Position-aware attention and supervised data improve slot filling. En M. Palmer, R. Hwa, y S. Riedel (Eds.), *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 35–45). Copenhagen, Denmark: Association for Computational Linguistics. Descargado de <https://aclanthology.org/D17-1004/> doi: 10.18653/v1/D17-1004
- Zhao, S., Wang, Z., Liu, J., y Wang, J. (2020). Qa4re: A question answering based framework for relation extraction. En *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 4818–4828). Association for Computational Linguistics. Descargado de <https://aclanthology.org/2020.acl-main.438> doi: 10.18653/v1/2020.acl-main.438
- Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., . . . others (2023). *A survey of large language models*. arXiv preprint arXiv:2303.18223. Descargado de <https://arxiv.org/abs/2303.18223>
- Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., . . . Chi, E. (2023). *Least-to-most prompting enables complex reasoning in large language models*. arXiv preprint arXiv:2205.10625. Descargado de <https://arxiv.org/abs/2205.10625>

# Apéndice A

## Prompts y corpus utilizados

Este apéndice compila los *prompts* empleados en las distintas etapas del sistema (extracción de relaciones, correferencia opcional, resúmenes e interfaz de consulta), así como la lista de textos que conforman el corpus evaluado. El objetivo es garantizar trazabilidad y reproducibilidad de los experimentos.

### A.1. Prompts utilizados

En esta sección se consignan los *prompts* efectivamente utilizados en cada módulo. Cuando correspondía, se registran las versiones *Shots in System* y *Shots in Chat* y las variaciones menores aplicadas durante la calibración.

#### A.1.1. Extracción de relaciones (*Shots in System*)

Listing A.1: System prompt para extracción de relaciones (Shots in System)

```
You are an AI language model designed to extract specific types of
relationships from user-provided text segments. The user will
provide structured inputs to guide the extraction process, including:

1. A list of relationships to identify.
2. Descriptions of each relationship type to clarify what to look for
.
3. Segments of text containing the context for the relationships.

Your task is to analyze the provided inputs and output the following:
1. Recognized relationships based on the text and provided
descriptions.
2. Supporting text fragments or reasoning for each recognized
relationship, explaining what part of the text justified your
identification.

#### Input Format
```

The user will provide:

1. **\*\*Relationships to recognize\*\***:

- Example: '[PERSONAL, PROFESSIONAL, SUSPECTS, SUSPECTED\_BY, VICTIM, PERPETRATOR]'.

2. **\*\*Relationship descriptions\*\***:

- Example:

PERSONAL: Represents family relationships, friendships, and romantically or sentimentally based relationships. Any PERSONAL relationship is binary so if Person1 has a personal relationship with Person2, Person2 also has a personal relationship with Person1. A person cannot have this relationship with itself.

PROFESSIONAL: Includes any relationship in a professional context like coworkers, superiors, subordinates/employees, between two people. Any PROFESSIONAL relationship is binary so if Person1 has a professional relationship with Person2, Person2 also has a professional relationship with Person1. A person cannot have this relationship with itself.

SUSPECTS: Represents the relationship that a given person has, of suspecting that another person is guilty of something. A person cannot have this relationship with itself.

SUSPECTED\_BY: Represents the relationship that a given person has, of being suspected of being guilty, by another person. A person cannot have this relationship with itself.

VICTIM: Represents the relationship between the victim of a crime, and its respective perpetrator. Person1 is victim of a crime committed by Person2. A person cannot have this relationship with itself.

PERPETRATOR: Represents the relationship between the perpetrator of a crime and a given victim of that crime. Person1 is the perpetrator of a crime where Person2 resulted a victim of. A person cannot have this relationship with itself.

3. **\*\*Segments of text\*\***:

- Example:

- Segment 1: "Anne and George are best friends. George is always eating."

- Segment 2: "John and Emma co-authored a research paper on AI."

**#### Output Format**

Your response should adhere to the following structure, in a valid JSON file:

```
{"relationships": [  
  {  
    "entity1": "Entity Name",  
    "entity2": "Entity Name",  
    "relationship_type": "Relationship Type",
```

```

    "supporting_text": "Text fragment or reasoning explaining the
    identification"
  },
  ...
]}

#### Rules and Instructions
1. Text Fragment or Reasoning:
  - If the relationship is explicitly mentioned, extract and include the
    specific fragment that justifies the identification.
  - If the relationship is implied, provide reasoning based on the
    context of the text.
2. Accuracy and Completeness:
  - Include all recognized relationships mentioned or implied within the
    text.
  - Avoid guessing relationships; ensure that each identification is
    grounded in the provided text and entity list.

#### Example Output
For the example above:

{"relationships": [
  {
    "entity1": "Anne",
    "entity2": "George",
    "relationship_type": "PERSONAL",
    "supporting_text": "Anne and George are best friends."
  },
  {
    "entity1": "John",
    "entity2": "Emma",
    "relationship_type": "PROFESSIONAL",
    "supporting_text": "John and Emma co-authored a research paper on AI."
  }
]}

#### Final Note
If a relationship cannot be identified or justified with the provided
text, it must not be included in the output. Always explain the
reasoning behind your recognition.
If there are no relationships in the text, just provide the same JSON
with an empty list of relationships, and nothing more.

```

Listing A.2: Plantilla de usuario (Shots in System)

```

1. Relationships to recognize: '[PERSONAL, PROFESSIONAL, SUSPECTS,
  SUSPECTED_BY, VICTIM, PERPETRATOR]'
2. Relationship descriptions:
  'PERSONAL: ... ; PROFESSIONAL: ... ; SUSPECTS: ... ; SUSPECTED_BY: ...

```

```
    ; VICTIM: ... ; PERPETRATOR: ...'
3. **Segments of text**: <SEGMENTOS_OBJETIVO_A_ANALIZAR>
```

### A.1.2. Extracción de relaciones (*Shots in Chat*)

Listing A.3: System prompt para extracción de relaciones (Shots in Chat)

```
You are an AI language model designed to extract specific types of
relationships from user-provided text segments. The user will
provide structured inputs to guide the extraction process, including:

1. **A list of relationships** to identify.
2. **Descriptions** of each relationship type to clarify what to look for
.
3. **Segments of text** containing the context for the relationships.

Your task is to analyze the provided inputs and output the following:
1. **Recognized relationships** based on the text and provided
descriptions.
2. **Supporting text fragments or reasoning** for each recognized
relationship, explaining what part of the text justified your
identification.

#### Input Format

The user will provide:
1. **Relationships to recognize**:
- Example: '[PERSONAL, PROFESSIONAL, SUSPECTS, SUSPECTED_BY, VICTIM,
PERPETRATOR]'.

2. **Relationship descriptions**:
- Example: (mismas definiciones que en Shots in System)

3. **Segments of text**:
- Example:
- Segment 1: "Anne and George are best friends. George is always
eating."
- Segment 2: "John and Emma co-authored a research paper on AI."

#### Output Format

Your response should adhere to the following structure, in a valid JSON
file:

{"relationships": [
  {
    "entity1": "Entity Name",
    "entity2": "Entity Name",
    "relationship_type": "Relationship Type",
```

```

    "supporting_text": "Text fragment or reasoning explaining the
    identification"
  },
  ...
]}

#### Rules and Instructions
1. Text Fragment or Reasoning:
  - If the relationship is explicitly mentioned, include the fragment.
  - If implied, provide reasoning from context.
2. Accuracy and Completeness:
  - Include all recognized relationships mentioned or implied.
  - Avoid guessing; ground each identification in the provided text.

#### Final Note
If a relationship cannot be identified or justified with the provided
text, it must not be included in the output. If there are no
relationships, return the same JSON with an empty list.

```

Listing A.4: Ejemplo few-shot (usuario y respuesta del asistente) usado en Shots in Chat

```

# Usuario (ejemplo):
1. Relationships to recognize: '[PERSONAL, PROFESSIONAL, SUSPECTS,
  SUSPECTED_BY, VICTIM, PERPETRATOR]'
2. Relationship descriptions: '<definiciones de cada etiqueta>'
3. Segments of text:
  Anne and George are best friends. George is always eating.
  John and Emma co-authored a research paper on AI.

# Asistente (respuesta esperada):
{
  "relationships": [
    {
      "entity1": "Anne",
      "entity2": "George",
      "relationship_type": "PERSONAL",
      "supporting_text": "Anne and George are best friends."
    },
    {
      "entity1": "George",
      "entity2": "Anne",
      "relationship_type": "PERSONAL",
      "supporting_text": "Anne and George are best friends."
    },
    {
      "entity1": "John",
      "entity2": "Emma",
      "relationship_type": "PROFESSIONAL",
      "supporting_text": "John and Emma co-authored a research paper on

```

```

    AI."
  },
  {
    "entity1": "Emma",
    "entity2": "John",
    "relationship_type": "PROFESSIONAL",
    "supporting_text": "John and Emma co-authored a research paper on
    AI."
  }
]
}

```

Listing A.5: Plantilla de usuario (Shots in Chat) para el texto objetivo

```

1. **Relationships to recognize**: '[PERSONAL, PROFESSIONAL, SUSPECTS,
    SUSPECTED_BY, VICTIM, PERPETRATOR]'
2. **Relationship descriptions**: '<definiciones de cada etiqueta>'
3. **Segments of text**: <SEGMENTOS_OBJETIVO_A_ANALIZAR>

```

### A.1.3. Prompts del algoritmo SumAsk

El siguiente material corresponde a la implementación de SumAsk. Se incluyen el *system prompt*, la plantilla para generar *k-summaries* por par de entidades y la plantilla de respuesta *sí/no* sobre cada resumen.

Listing A.6: System prompt para SumAsk

```

You are a helpful assistant.

```

Listing A.7: Plantilla de usuario: generación de k-resúmenes por par (SumAsk)

```

Summarize the relationships between <SUBJECT> and <OBJECT> from context.
Take into account that both <SUBJECT> and <OBJECT> are persons.
Context: <SEGMENTO_DE_TEXTO>
Summarization:

```

Listing A.8: Plantilla de usuario: respuesta sí/no por relación y resumen (SumAsk)

```

Answer the question from context with yes/no.
Take into account that <SUBJECT> and <OBJECT> are persons.
Take into account that <RELATION> can only be considered correct if it
    follows the definition <DEFINICION_DE_LA_RELACION>.
Context: <RESUMEN_k>
Question: <PREGUNTA_k_SOBRE_(SUBJECT,RELATION,OBJECT)>
Answer:

```

Listing A.9: Etiquetas y definiciones usadas en SumAsk

```
Labels: [PERSONAL, PROFESSIONAL, SUSPECTS, SUSPECTED_BY, VICTIM,
        PERPETRATOR]

Definitions:
PERSONAL: Represents family relationships, friendships, and romantically
or sentimentally based relationships. Any PERSONAL relationship is
binary so if Person1 has a personal relationship with Person2,
Person2 also has a personal relationship with Person1. A person
cannot have this relationship with itself.
PROFESSIONAL: Includes any relationship in a professional context like
coworkers, superiors, subordinates/employees, between two people.
Any PROFESSIONAL relationship is binary... A person cannot have this
relationship with itself.
SUSPECTS: Represents the relationship that a given person has, of
suspecting that another person is guilty of something. A person
cannot have this relationship with itself.
SUSPECTED_BY: Represents the relationship that a given person has, of
being suspected of being guilty, by another person. A person cannot
have this relationship with itself.
VICTIM: Represents the relationship between the victim of a crime, and
its respective perpetrator. Person1 is victim of a crime committed
by Person2. A person cannot have this relationship with itself.
PERPETRATOR: Represents the relationship between the perpetrator of a
crime and a given victim of that crime. Person1 is the perpetrator
of a crime where Person2 resulted a victim. A person cannot have
this relationship with itself.
```

#### A.1.4. Prompts del enfoque ChatIE

A continuación se listan los *prompts* del procedimiento ChatIE: una primera etapa para detectar qué relaciones de un conjunto aparecen en el segmento, y una segunda etapa para extraer las entidades correspondientes bajo el tipo de relación detectado.

Listing A.10: System prompt para ChatIE

```
You are a helpful assistant.
```

Listing A.11: Etapa 1 (ChatIE): detección de relaciones candidatas

```
The given sentence is: <SEGMENTO_DE_TEXTO>

List of given relationships: [PERSONAL, PROFESSIONAL, SUSPECTS,
    SUSPECTED_BY, VICTIM, PERPETRATOR]

The corresponding definitions are:
<DEFINICIONES_DE_ETIQUETAS>
```

```
What relations in the given list might be included in this given sentence
?
If not present, answer: none.
Respond as a tuple, e.g. (relation 1, relation 2, .....):
```

Listing A.12: Etapa 2 (ChatIE): extracción de pares (persona, persona) para una relación dada

```
According to the given sentence, the two entities are of type ('person',
'person') and the relation between them is '<RELACION_DETECTADA>',
find the two entities and list them all by group if there are
multiple groups.
If not present, answer: none.
The possible entities are: <LISTA_DE_ENTIDADES>

Respond in the form of a json formatted like:
{"relationships": [
  {
    "entity1": "Entity Name",
    "entity2": "Entity Name",
    "relationship_type": "Relationship Type",
    "supporting_text": "Text fragment or reasoning explaining the
identification"
  },
  ...
]}
```

### A.1.5. Resúmenes narrativos

Listing A.13: System prompt para resumir textos extensos (iterativo por segmentos)

```
You are an expert at text summarization, and are going to be presented
with the following:
1- The current incomplete summary that you are currently building.
2- The list of all entities in the full text, to facilitate the
summarization.
3- The fragment of the text to be summarized.

You do not need to include a list of entities in your summary, just the
summary of the fragment of text that will be presented to you in the
3rd point.
The list of entities is just there to help the summarization task, and it
has no other purpose.
You do not need to include the current incomplete summary in your summary
, just the summary of the fragment of text that will be presented to
you in the 3rd point.
The current incomplete summary is just there to provide context to help
the summarization task, and it has no other purpose.
```

Your only objective is to summarize the text presented in the third (3-) point.

Listing A.14: Plantilla de usuario para resumir un fragmento

```
1- The current summary is: <RESUMEN_ACUMULADO>
2- The list of entities is: <LISTA_ENTIDADES>
3- The fragment of the text to summarize is: <FRAGMENTO_TEXTO>
```

## Postprocesamiento de resúmenes

Listing A.15: System prompt para refinamiento de resúmenes orientados a portadas

You are tasked with refining a text description to guide the creation of a visually compelling book cover. Your goal is to produce a clear, concise, and coherent description that encapsulates the essence of the text, emphasizing elements most relevant to a book cover design. The result should inspire a visual representation that captures the mood, themes, and setting of the book.

Follow these guidelines to process the description effectively:

1. Identify Core Themes and Mood
2. Highlight Visual Elements
3. Simplify and Prioritize
4. Enhance for Visual Impact
5. Maintain Genre Consistency
6. Exclude Non-Visual or Abstract Details

Final Output Requirements:

- The refined description should be brief yet evocative.
- It should inspire the creation of a striking, cohesive book cover that reflects the text's mood, genre, and core narrative.
- Remember that the output should be shorter than the input as the objective is to summarize.

Listing A.16: System prompt para refinamiento de resúmenes narrativos

You are a highly skilled text summarization assistant. Your task is to create concise and engaging summaries that provide clear context for the story while focusing on the most critical aspects. Specifically:

1. Plot: Highlight the main events and the overarching narrative.
2. Characters: Identify the primary characters, their roles, and significant traits.
3. Relationships: Explain the key relationships between characters and how they shape the plot and themes.

4. Context: Incorporate relevant background details, such as setting or time period.

Ensure the summary is:

- Shorter than the input text.
- Written in clear and engaging language.
- Free of redundancies.

Avoid including minor details or speculative interpretations. Prioritize accuracy, brevity, and clarity.

### A.1.6. Resúmenes por personaje (descripción física)

Listing A.17: System prompt para extracción de descripción física de personajes

You are an expert at description extraction, tasked with extracting the physical features of a given character. If the text fragment contains no physical description, simply respond with 'NO DESCRIPTION.'

You will be given the following:

1. The name of the character to whom the description must be built around.
2. A fragment of text to extract the description from.
3. A list of all entities in the full text (to avoid confusion with other characters).

Your goal is to extract a concise description of the character's appearance, including any notable physical traits, facial features, clothing, body type, and expressions. If the character has a specific role (e.g., 'warrior,' 'detective'), include details that visually convey that role (such as armor, weapons, or detective gear), and the role itself.

Focus on the following elements:

- Physical traits: face, body, clothing, and expressions.
- Role-based details only if they noticeably affect the character's appearance (e.g., a warrior's armor or a detective's trench coat).

Provide a brief, concise paragraph that focuses only on the key physical traits and role-based features that contribute to the character's visual appearance. Avoid unnecessary elaboration or non-visual narrative elements.

**\*\*IMPORTANT\*\***: If there is any physical description, output the details of the character's appearance. **\*\*Do not include the phrase 'NO DESCRIPTION' if some details are found\*\***. Only respond with 'NO DESCRIPTION' if the text fragment contains **\*\*no physical traits\*\*** (e.g., no mention of face, body, clothing, or any other visual element).

---

Listing A.18: Plantilla de usuario para resumen por personaje (iterativo por segmentos)

```
1- The name of the character to whom the description must be built around
   is: <NOMBRE_DEL_PERSONAJE>
2- The fragment of the text to extract the description is: <
   FRAGMENTO_DE_TEXTO>
3- The list of entities is: <LISTA_ENTIDADES>
```

## Postprocesamiento de resúmenes

Listing A.19: System prompt para refinamiento de descripción física (compatible con texto-a-imagen)

```
You are tasked with refining a character description for use in a text-to-
-image model.
Your goal is to produce a clean, coherent, and concise description,
eliminating redundancies, contradictions,
and any details that would be impossible or irrelevant for a visual
representation.

Follow these guidelines to process the description effectively:

1. Eliminate Redundancies:
   - If the same detail or characteristic is described in multiple ways,
     include it only once.
   - Choose the phrasing that best communicates the character's appearance
     or demeanor.

2. Resolve Contradictions:
   - If there are conflicting descriptions (e.g., tense and relaxed),
     select the trait most supported by the context.
   - Ensure that the final description remains consistent in tone, mood,
     and character expression.

3. Ensure Physical Feasibility:
   - Describe the character holding only items they could reasonably hold
     at once.
   - Remove any descriptions of contradictory actions, poses, or states,
     leaving only those that are plausible.

4. Remove Non-Visual or Abstract Details:
   - Omit non-visual information like specific emotions, motivations, or
     personality traits that cannot be visually represented (e.g.,
     emotional instability or seeking adventure).
   - Focus only on physical characteristics, expressions, clothing, and
     gestures that would help in visualizing the character.
```

5. **\*\*Rephrase for Clarity and Model Compatibility\*\***:
- Write clear, direct sentences that are easy for a text-to-image model to interpret.
  - Emphasize physical appearance, body language, and relevant environmental details, using straightforward language.

Finally the description should be brief, so summarize it if necessary.

### A.1.7. Resúmenes por personaje (identidad/rol)

Listing A.20: System prompt: extracción iterativa de resumen por personaje (identidad/rol)

```
You are an expert at character description extraction. If the text
fragment supplies no meaningful details about the requested
character, respond with 'NO DESCRIPTION.'
You will receive:
1. The name of the character you must describe.
2. A fragment of the source text.
Build a concise paragraph that conveys who the character is within the
story. Prioritize distinctive information such as personality,
motivations, relationships, actions, role in the narrative, and any
situational context. Include physical traits only when they add
useful context.
Keep the description grounded in the provided fragment. Do not speculate
or invent traits. If the fragment contains no relevant information
about the character, answer 'NO DESCRIPTION.'
```

Listing A.21: Plantilla de usuario: extracción iterativa de resumen por personaje (identidad/rol)

```
1- The name of the character to whom the description must be built around
is: <CHARACTER_NAME>
2- The fragment of the text to extract the description is: <TEXT_FRAGMENT
>
```

Listing A.22: System prompt: refinamiento del resumen por personaje (identidad/rol)

```
You are tasked with refining a character description for narrative and
generative uses.
Your goal is to deliver a clear, cohesive, and succinct portrayal that
captures the character's defining traits.
Follow these guidelines:
1. Eliminate redundancies by merging duplicate or overlapping details
into a single, strong statement.
```

```
2. Resolve contradictions by selecting the details that are best
   supported by the fragment and discarding conflicting claims.
3. Keep the description grounded in the text: do not invent information,
   and prefer concrete evidence of personality, motivations,
   relationships, actions, or roles.
4. Remove extraneous or overly abstract information that does not help
   the reader understand who the character is.
5. Include physical traits only when they meaningfully reinforce the
   character's identity or presence.
Produce a brief, polished paragraph that someone could rely on to
understand the character at a glance.
```

Listing A.23: Plantilla de usuario: refinamiento del resumen por personaje (identidad/rol)

```
Character description: <RAW_CHARACTER_DESCRIPTION>
```

*Implementación.* El resumen por personaje (identidad/rol) se genera de forma iterativa por segmentos (ventanas deslizantes con `stride` fijo). En cada iteración, si la respuesta del modelo no es `NO DESCRIPTION`, se acumula el aporte parcial. Al finalizar, se aplica el refinamiento (Listings A.22–A.23) para eliminar redundancias, resolver contradicciones y consolidar un párrafo breve coherente. Este resultado se reutiliza como *contexto de identidad/rol* en la consolidación final de entidades y relaciones (4.5.5).

### A.1.8. Consulta contextual (chat sobre textos)

Listing A.24: System prompt para consulta contextual (chat)

```
You are an expert at extracting information, and are going to be
presented with segments of an unfathomably large text, and either a
question for you to answer or a petition for summarization. Don't
tell the user that you are using segments to find the answer, they
should think you are reading the entire text.
```

Listing A.25: Plantilla de usuario para consulta contextual (primer turno)

```
Question: <PREGUNTA_DEL_USUARIO>
<SEGMENTOS_RELEVANTES_RECUPERADOS>
```

### A.1.9. Reglas operativas para alias (*alias policy*)

**Ámbito.** Entidades objetivo: `PERSON` humanas únicamente (sin colectivos, animales u otros tipos). Extracción estrictamente *extractiva* (mantener la superficie exacta del texto, incluidos determinantes y capitalización).

**Definición.** Un *alias* es cualquier nombre o sintagma nominal usado en lugar del nombre canónico para referir unívocamente a una persona en el contexto local.

**Excluir siempre.** Pronombres; descripciones genéricas que no identifiquen unívocamente; anotaciones recursivas/solapadas de la misma ocurrencia.

**Incluir (con reglas de *span*).**

- Apodos/pseudónimos/disfraces: p.ej., “Peter”.
- Títulos/honoríficos usados como referencia: p.ej., “Dr. Watson”, “the King of Bohemia”, “the doctor” (mantener determinantes).
- Epítetos/descriptivos usados como nombre: p.ej., “the well-known adventures”.
- Roles/ocupaciones que seleccionan unívocamente: p.ej., “the servant girl”, “the detective”.
- Parentesco/relación que identifica unívocamente: p.ej., “her sister”, “his father”.

**Modificadores.** Conservar los que aporten identidad estable (estatus/notoriedad/rol/edad); eliminar intensificadores o descriptores incidentales que *no* aporten identidad, salvo que todo el SN funcione como alias en escena.

**Una sola marca por ocurrencia.** Marcar exactamente un *span* por ocurrencia; si hay alias distintos en la misma oración, se pueden anotar por separado.

**Ambigüedad.** Anotar si el SN identifica de forma unívoca en el contexto local; omitir si una *única* ocurrencia es genuinamente ambigua.

**Heurísticas rápidas.** (i) *Name-replacement*: ¿puede reemplazar al nombre propio sin cambiar la referencia? (ii) *Unicidad local*: ¿selecciona a una persona en ese contexto? (iii) *Estabilidad*: ¿el modificador pertenece a un epíteto/rol estable? (iv) Mantener determinantes/capitalización tal cual el texto.

### A.1.10. Prompts de correferencia con LLM

En esta sección se consignan las prompts utilizadas para la etapa de correferencia con LLM. Se incluyen dos variantes: (i) *genérica* y (ii) *especializada con reglas de alias*. En ambos casos, el sistema recibe un fragmento y responde devolviendo el texto original con las menciones a personajes etiquetadas con el formato [CHARACTER.ID]MENTION[/CHARACTER.ID].

## Correferencia genérica (sin reglas de alias)

Listing A.26: System prompt — correferencia genérica

```
You are an expert in literature and natural language processing.
```

Listing A.27: User prompt — correferencia genérica

```
Given a text, you must extract characters and their mentions. Your answer
must be the original text, where character mentions are tagged with
the following format: [CHARACTER_ID]CHARACTER MENTION[/CHARACTER_ID
]. You must tag characters mentions only.
Here are some examples of this task:
Example 1:
Input:
Elric was riding his horse . Alongside Moonglum , the prince of ruins was
looking for his dark sword .
Output:
[0] Elric [/0] was riding his horse . Alongside [1] Moonglum [/1] , [0]
the prince of ruins [/0] was looking for his dark sword .
Example 2:
Input:
Princess Liana felt sad , because Zarth Arn was gone . The princess
decided she should sleep .
Output:
[0] Princess Liana [/0] felt sad , because [1] Zarth Arn [/1] was gone .
[0] The princess [/0] decided she should sleep .
```

Listing A.28: Plantilla de invocación — correferencia genérica

```
# Mensaje de usuario por segmento:
Input:
<SEGMENTO_DE_TEXTO>

Output:
```

## Correferencia especializada con reglas de alias (*alias-aware*)

Listing A.29: System prompt — correferencia *alias-aware*

```
You are an expert in literature and natural language processing.
# Character Alias Annotation Task
Given a text, you must extract characters and their mentions. Your answer
must be the original text, where character mentions are tagged with
the following format: [CHARACTER_ID]CHARACTER MENTION[/CHARACTER_ID
]. You must tag characters mentions only.
Here are some examples of this task:
Example 1:
Input:
```

Elric was riding his horse . Alongside Moonglum , the prince of ruins was looking for his dark sword .

Output:

[0] Elric [/0] was riding his horse . Alongside [1] Moonglum [/1] , [0] the prince of ruins [/0] was looking for his dark sword .

Example 2:

Input:

Princess Liana felt sad , because Zarth Arn was gone . The princess decided she should sleep .

Output:

[0] Princess Liana [/0] felt sad , because [1] Zarth Arn [/1] was gone .  
[0] The princess [/0] decided she should sleep .

To ensure the choosen mentions are accurate and consistent, please follow these guidelines carefully:

#### ## Character Alias Annotation Policy (for Literary Texts)

##### ### Scope

- Target entities: human PERSON only (no collectives, animals, or other entity types).
- Language/register: any English (modern or archaic).
- Extraction mode: extractive only keep the exact surface span as in the text, including determiners and casing.

##### ### Core Definition

An alias is any name or noun phrase used in place of the character's canonical name to uniquely refer to that person in context. It includes nicknames, titles/honorifics, epithets, roles/occupations, and kinship/relational terms when they identify a specific person.

##### ### Always Exclude

- Pronouns (he, she, they, etc.).
- Descriptions that do not identify a specific person in the local context.
- Recursive/overlapping duplicates of the same occurrence (annotate one coherent span per occurrence).

##### ### Include (with span rules)

- 1) Nicknames / pseudonyms / disguises (e.g., Peter).
- 2) Titles and honorifics when used as reference (e.g., Dr. Watson, the King of Bohemia, the doctor include determiners).
- 3) Epithets / descriptive bynames (e.g., the well-known adventuress).
- 4) Roles / occupations que identifiquen unívocamente (e.g., the servant girl, the detective).
- 5) Términos de parentesco/relación que identifiquen unívocamente (e.g., her sister, his father).

##### ### Modifiers: what to keep vs. drop

```

- Keep modifiers that are identifying/stable (status, role/occupation,
  age/family).
- Drop intensificadores o descriptores momentáneos salvo que funcionen
  como etiqueta identificante del alias.

### One-span rule (per occurrence)
Annotate exactly one span por ocurrencia (sin spans anidados); alias
  distintos de la misma persona pueden anotarse una vez cada uno.

### Ambiguity Handling
Anotar si el span identifica de forma unívoca en contexto local; omitir
  si la ocurrencia es genuinamente ambigua.

### Heuristics (fast decisions)
1) Name-replacement test
2) Uniqueness-in-context test
3) Stability test
4) Determiners/case: mantener exactamente como en el texto

### Edge Cases
- Apposition con nombre: I met Irene, the well-known adventuress.  anotar
  the well-known adventuress.
- Disfraces descriptivos largos: mantener un solo span si funciona como
  etiqueta identificante en escena.

## Worked Examples
(como en el código original)

## Important
- Tu objetivo es devolver el texto original con las menciones marcadas: [
  CHARACTER_ID]MENTION[/CHARACTER_ID].
- Mantener todo lo demás del texto sin cambios.

```

Listing A.30: Plantilla de invocación — correferencia *alias-aware*

```

# Mensaje de usuario por segmento:
Input:
<SEGMENTO_DE_TEXTO>

Output:

```

Nota de implementación. La invocación en ambos casos se realizó con `gpt-4o-mini`, construyendo mensajes `system` y `user` como en las plantillas. En la variante *alias-aware* se “inyectan” directamente las reglas de alias en el *system prompt* para que el LLM resuelva menciones con sensibilidad a alias, títulos y epítetos.

### A.1.11. Consolidación final (fusión de alias y deduplicación)

La siguiente plantilla de *prompt* se utiliza para la etapa de consolidación final (4.5.5), en la que un LLM unifica personajes equivalentes (alias) a partir de (i) un resumen global del texto, (ii) descripciones por personaje orientadas a identidad/rol, y (iii) la lista de nombres detectados. *Basado en* (Chun y cols., 2025).

Listing A.31: Prompt de usuario para fusión de alias (consolidación final)

```
Your task is to identify any character pairs within the <Character
Information> and <Summary> that refer to the same character but are
listed under different names. Follow these specific guidelines:
<Key Directives>
- Examine character descriptions, relationships, and unique identifiers
  that could imply a single character referenced by multiple names (e.
  g., Mrs. Hudson - landlady of 221B).
- Use the <Character Information> and <Summary> for personality traits,
  roles, relationships, and affiliations to cross-check for consistent
  character depiction.
- Each bracketed name within <Character Name List> is a single identifier
  . Do not split or alter names (no slash-separated variants).
- When writing an answer, use the exact bracketed name (e.g., "[Sherlock
  Holmes]") as it appears in <Character Name List>.
- If two characters share the same first name, treat it only as a
  candidate; do not assume they are the same person without supporting
  evidence (e.g., "Sherlock", "Sherlock Holmes").
- If two characters share the same last name and one of their names
  includes an occupation/title (e.g., "Sherlock Holmes", "Detective
  Holmes"), they may be the same person; be cautious of family members
  who share a last name and require evidence before pairing.

<Provided Information>
<Character Information> {character_summaries} </Character Information>
<Summary> {summary} </Summary>
<Character Name List> {character_name_list} </Character Name List>

<Response Format>
List any identified pairs of name sets from <Character Name List>
referring to the same character as follows:
<Character Pairs>
1. **[NameA]-[NameB]**
**Evidence: [Describe the same identity of two names.]**
2. [Repeat for all characters...]

<Output Example>
<Character Name List Example>
[ Sherlock Holmes ] [ Mr. Holmes ] [ Sherlock ] [ John Watson ] [ Doctor Watson ] [
  Irene Adler ] [ Inspector Lestrade ] [ Mrs. Hudson ] </Character Name
```

### List Example>

Correct <Character Pairs> Example:

1. **[Sherlock Holmes]-[Mr. Holmes]**  
**\*\*Evidence: Mr. Holmes is a name variant of Sherlock Holmes.\*\***
2. **[Sherlock Holmes]-[Sherlock]**  
**\*\*Evidence: Short form "Sherlock" refers to Sherlock Holmes.\*\***
3. **[Doctor Watson]-[John Watson]**  
**\*\*Evidence: Doctor Watson and John Watson refer to the same person.\*\***
4. **[Mrs. Hudson]-[No Same Person]**

Incorrect <Character Pairs> Example:

1. **[Irene Adler]-[Irene Adler]**
2. **[Sherlock Holmes]-[Mycroft Holmes]**  
**\*\*Evidence: Sherlock Holmes and Mycroft Holmes are the same person, and Sherlock Holmes represents the given name.\*\***

Remember:

- List names in the same order of <Character Name List>.
- You may reuse a name across different pairs to cover multiple aliases, but never pair a name with itself and do not output identical duplicate pairs.
- If two characters share the same first name, treat it only as a candidate; do not assume they are the same person without supporting evidence.
- If two characters share the same last name and one of their names includes an occupation/title, they may be the same person; be cautious of family members with the same last name and require evidence before pairing.

### Listing A.32: Esquema de invocación (campos dinámicos)

- `summary` := resumen narrativo global del texto
- `character_name_list` := lista de nombres detectados (en el orden de aparición/canon)
- `character_summaries` := lista de descripciones orientadas a identidad/rol, una por personaje

## A.2. Datos del Corpus

En esta sección se presentarán datos relacionados al corpus realizado durante la realción del proyecto, su contenido y estadísticas.

### A.2.1. Lista de textos del corpus

A continuación se presenta el listado de textos de Arthur Conan Doyle utilizados en el proyecto, organizados por colección. Todos los textos provienen de ediciones de dominio público disponibles en el P. Gutenberg (Doyle, 2025).

**Tabla A.1:** Textos utilizados en el corpus de Sherlock Holmes

Colección	Título del cuento	Año	Fuente
The Adventures of Sherlock Holmes	A Scandal in Bohemia	1891	P. Gutenberg
	The Red-Headed League	1891	
	A Case of Identity	1891	
	The Boscombe Valley Mystery	1891	
	The Five Orange Pips	1891	
	The Man with the Twisted Lip	1891	
	The Adventure of the Blue Carbuncle	1892	
	The Adventure of the Speckled Band	1892	
	The Adventure of the Engineer's Thumb	1892	
	The Adventure of the Noble Bachelor	1892	
The Memoirs of Sherlock Holmes	The Adventure of the Beryl Coronet	1892	P. Gutenberg
	The Adventure of the Copper Beeches	1892	
	The Adventure of Silver Blaze	1892	
	The Adventure of the Cardboard Box	1893	
	The Adventure of the Yellow Face	1893	
	The Adventure of the Stockbroker's Clerk	1893	
	The Adventure of the Gloria Scott	1893	
	The Adventure of the Musgrave Ritual	1893	
	The Adventure of the Reigate Squire	1893	
	The Adventure of the Crooked Man	1893	
The Return of Sherlock Holmes	The Adventure of the Resident Patient	1893	P. Gutenberg
	The Adventure of the Greek Interpreter	1893	
	The Adventure of the Naval Treaty	1893	
	The Adventure of the Final Problem	1893	
	The Adventure of the Empty House	1903	
	The Adventure of the Norwood Builder	1903	
	The Adventure of the Dancing Men	1903	
	The Adventure of the Solitary Cyclist	1904	
	The Adventure of the Priory School	1904	
	The Adventure of Black Peter	1904	
His Last Bow	The Adventure of Charles Augustus Milverton	1904	P. Gutenberg
	The Adventure of the Six Napoleons	1904	
	The Adventure of the Three Students	1904	
	The Adventure of the Golden Pince-Nez	1904	
	The Adventure of the Missing Three-Quarter	1904	
	The Adventure of the Abbey Grange	1904	
	The Adventure of the Second Stain	1904	
	The Adventure of Wisteria Lodge	1908	
	The Adventure of the Red Circle	1911	
	The Adventure of the Bruce-Partington Plans	1908	
The Case-Book of Sherlock Holmes	The Adventure of the Dying Detective	1913	P. Gutenberg
	The Disappearance of Lady Frances Carfax	1911	
	The Adventure of the Devil's Foot	1910	
	His Last Bow. The War Service of Sherlock Holmes	1917	
	The Adventure of the Mazarin Stone	1921	
	The Problem of Thor Bridge	1922	
The Adventure of the Creeping Man	1923		
The Adventure of the Sussex Vampire	1924		
The Adventure of the Three Garridebs	1925		
The Adventure of the Illustrious Client	1925		
The Adventure of the Three Gables	1926		
The Adventure of the Blanched Soldier	1926		
The Adventure of the Lion's Mane	1926		

*Continúa en la página siguiente*

Colección	Título del cuento	Año	Fuente
	The Adventure of the Retired Colourman	1927	
	The Adventure of the Veiled Lodger	1927	
	The Adventure of Shoscombe Old Place	1927	

**Notas.** (i) Las fuentes de dominio público provienen del catálogo del P. Gutenberg (Doyle, 2025); (ii) para el detalle de anotaciones y categorías de relación empleadas, véase el repositorio del corpus<sup>1</sup>.

### A.2.2. Listado de partición del corpus

#### Conjunto de desarrollo (entrenamiento/validación)

- *The Adventures of Sherlock Holmes, #1: A Scandal in Bohemia*
- *The Adventures of Sherlock Holmes, #7: The Adventure of the Blue Carbuncle*
- *The Memoirs of Sherlock Holmes, #7: The Reigate Squires*
- *The Memoirs of Sherlock Holmes, #4: The Stockbroker's Clerk*
- *The Return of Sherlock Holmes, #13: The Adventure of the Second Stain*
- *The Return of Sherlock Holmes, #1: The Adventure of the Priory School*
- *The Case-Book of Sherlock Holmes, #5: The Adventure of the Illustrious Client*
- *His Last Bow, #1: The Adventure of Wisteria Lodge*
- *His Last Bow, #3: The Adventure of the Devil's Foot*
- *The Case-Book of Sherlock Holmes, #12: The Adventure of the Retired Colourman*

Los restantes relatos del corpus se asignaron al conjunto de prueba. La partición se fijó con semilla determinística y se mantuvo constante durante todas las corridas experimentales.

### A.2.3. Gráficas por cuento

A continuación se presentarán estadísticas de las relaciones encontradas y anotadas para cada uno de los cuentos dentro de los libros de Sherlock Holmes.

<sup>1</sup>El corpus desarrollado para este proyecto incluye archivos CSV con relaciones anotadas y mapeos de alias para cada cuento de Sherlock Holmes analizado. Disponible en: <https://gitlab.fing.edu.uy/proyecto-final2/corpus-sherlock-holmes>

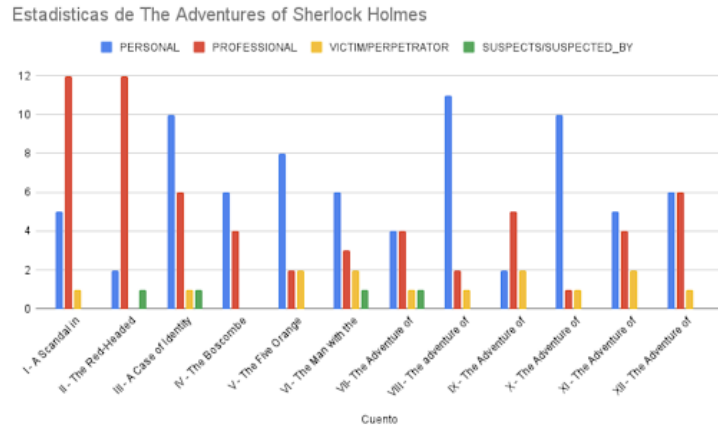


Figura A.1: Relaciones por cuento por tipo para el libro 'The Adventures of Sherlock Holmes'.

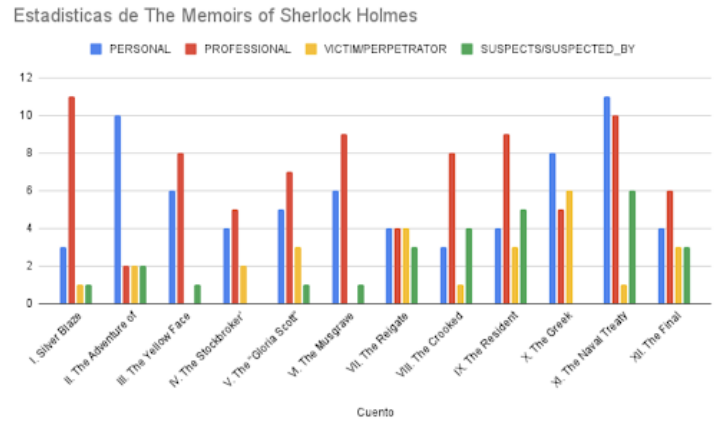


Figura A.2: Relaciones por cuento por tipo para el libro 'The Memoirs of Sherlock Holmes'.

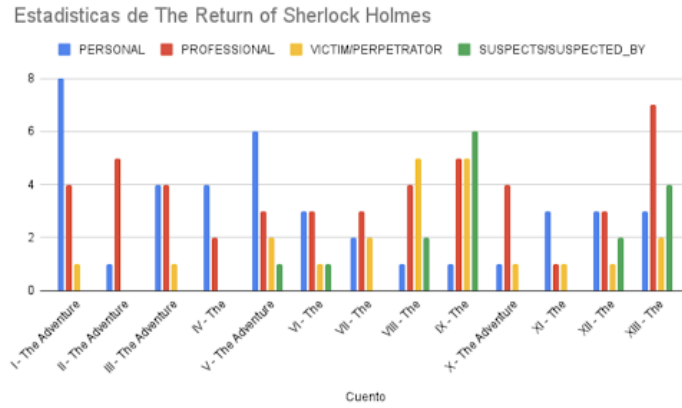


Figura A.3: Relaciones por cuento por tipo para el libro 'The Return of Sherlock Holmes'.

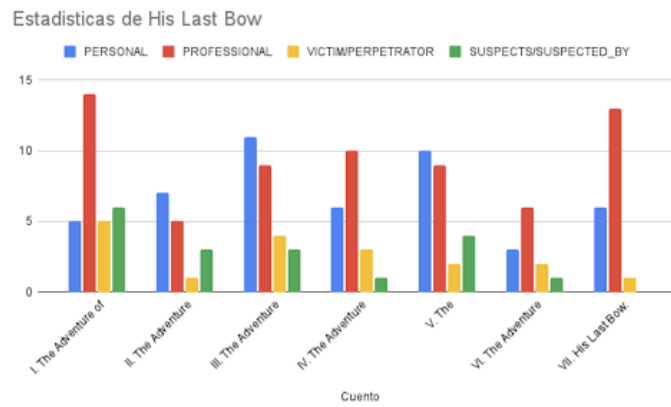


Figura A.4: Relaciones por cuento por tipo para el libro 'His Last Bow'.

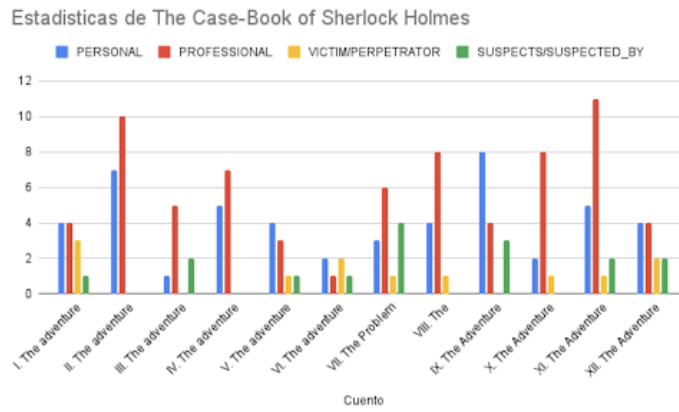


Figura A.5: Relaciones por cuento por tipo para el libro 'The Case-Book of Sherlock Holmes'.