

UNIVERSIDAD POLITÉCNICA DE MADRID  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE SISTEMAS  
INFORMÁTICOS



Universidad  
Politécnica  
de Madrid

ETSI **SISTEMAS  
INFORMÁTICOS**

**Alternativa Ligera a los LLMs en la Generación de  
Grafos de Conocimiento: Mejoras a partir de OpenIE**

Proyecto Fin de Grado

Grado en Ciencia de Datos e Inteligencia Artificial

Curso académico 2024-2025

Autor:  
Elia González de Heredia Solana

Tutor:  
Carlos Badenes Olmedo

# Resumen

Este proyecto propone un enfoque ligero para la generación de grafos de conocimiento a partir de texto, evitando el uso de grandes modelos de lenguaje. La base del sistema es un grafo inicial (Baseline) generado mediante OpenIE, que se enriquece a través de un conjunto de estrategias organizadas en tres bloques: expansión de cobertura, homogeneización y refinamiento, y mejora de precisión. Estas estrategias buscan mejorar la calidad del grafo en términos de precisión, cobertura y similitud estructural, evaluadas tanto en el plano léxico como en el semántico, respecto a un grafo de referencia (Gold).

Para evaluar el rendimiento del enfoque propuesto, se compara su resultado tanto con el grafo Gold elaborado manualmente, como con el grafo Baseline original. Los resultados muestran mejoras significativas en la mayoría de las métricas. En concreto, la precisión semántica se incrementa en un 96.8 %, la precisión léxica en un 70.6 %, y la similitud estructural con el grafo Gold mejora un 89.3 % en el plano semántico y un 60.4 % en el plano léxico. En cuanto a la cobertura, se observa un aumento del 15.4 % en el plano semántico, mientras que la cobertura léxica se mantiene constante. Aunque el sistema final no alcanza el rendimiento de los enfoques basados en modelos de lenguaje, los resultados obtenidos validan la efectividad del enfoque propuesto y confirman su capacidad para reducir de manera significativa la brecha de calidad frente a soluciones más complejas y costosas.

**Palabras clave:** Generación de grafos de conocimiento, OpenIE, Procesamiento del lenguaje natural, Enfoques ligeros

# Abstract

This project proposes a lightweight approach for generating knowledge graphs from text, avoiding the use of large language models. The system is based on an initial graph (Baseline) generated using OpenIE, which is enriched through a set of strategies organized into three blocks: coverage expansion, homogenization and refinement, and precision improvement. These strategies aim to enhance the quality of the graph in terms of precision, coverage, and structural similarity, all evaluated on both lexical and semantic levels with respect to a reference graph (Gold).

To assess the performance of the proposed approach, its results are compared to both the manually created Gold graph and the original Baseline graph. The results show significant improvements in most metrics. Specifically, semantic precision increases by 96.8 %, lexical precision by 70.6 %, and structural similarity with the Gold graph improves by 89.3 % in the semantic dimension and 60.4 % in the lexical one. Regarding coverage, there is a 15.4 % improvement in the semantic dimension, while lexical coverage remains unchanged. Although the final system does not match the performance of large language model-based approaches, the results validate the effectiveness of the proposed lightweight method and confirm its ability to significantly reduce the quality gap compared to more complex and resource-intensive solutions.

**Keywords:** Knowledge graph generation, OpenIE, Natural language processing, Lightweight approaches

# Índice

Resumen . . . . .	I
Abstract . . . . .	II
<b>1. Introducción</b>	<b>1</b>
1.1. Contexto . . . . .	1
1.2. Motivación . . . . .	3
1.3. Objetivos . . . . .	4
1.3.1. Objetivo general . . . . .	4
1.3.2. Objetivos específicos . . . . .	4
1.4. Estructura del documento . . . . .	4
<b>2. Estado del arte</b>	<b>6</b>
2.1. Construcción de KGs a partir de texto: Visión general . . . . .	6
2.2. Extracción de Información Abierta . . . . .	8
2.2.1. Desde TextRunner hasta los LLMs . . . . .	8
2.2.2. Proceso Básico OpenIE . . . . .	10
2.2.3. Desafíos de OpenIE . . . . .	11
2.3. Enfoques basados en LLMs . . . . .	12
2.4. Mejoras en la Calidad del Grafo de Conocimiento . . . . .	14
<b>3. Metodología</b>	<b>16</b>
3.1. Dataset de Referencia . . . . .	16
3.1.1. Objetivo del Dataset de Referencia . . . . .	16
3.1.2. Descripción del Dataset de Referencia . . . . .	16
3.2. Métricas de Evaluación . . . . .	18
3.2.1. Precisión . . . . .	19
3.2.2. Cobertura . . . . .	20
3.2.3. Similitud estructural . . . . .	20
3.3. Creación de Grafos de Conocimiento . . . . .	22
3.3.1. Grafo Gold . . . . .	22
3.3.2. Grafo Baseline con OpenIE . . . . .	22
3.3.3. Grafo Neo4j LLM Graph Builder . . . . .	23
3.3.4. Grafo Notebook LM . . . . .	24
3.4. Bloque I: Expansión de Cobertura . . . . .	25
3.4.1. Estrategia 1: Normalización por Chunking y Re-Extracción . . . . .	25
3.4.2. Estrategia 2: Extracción de Tripletas de Chunks con Verbos . . . . .	27
3.5. Bloque II: Homogeneización y Refinamiento . . . . .	28
3.5.1. Estrategia 3: Corrección de Verbos Modales . . . . .	28
3.5.2. Estrategia 4: Mapeo de Sujetos y Objetos a Chunks de CoreNLP . . . . .	29
3.5.3. Estrategia 5: Mapeo de Sujetos y Objetos con Chunks Limpiados y Enriquecidos (CoreNLP + SpaCy) . . . . .	30
3.6. Bloque III: Mejora de Precisión . . . . .	31
3.6.1. Estrategia 6: Filtrado de Tripletas por Confianza . . . . .	31
3.6.2. Estrategia 7: Eliminación de Tripletas Similares . . . . .	32

<b>ÍNDICE</b>	<b>IV</b>
<b>4. Resultados</b>	<b>33</b>
4.1. Evaluación Preliminar: Baseline frente a otros enfoques . . . . .	33
4.2. Resultados por Bloque de Estrategias . . . . .	38
4.2.1. Bloque I: Expansión de Cobertura . . . . .	38
4.2.2. Bloque II: Homogeneización y Refinamiento . . . . .	47
4.2.3. Bloque III: Mejora de Precisión . . . . .	58
4.3. Evaluación final: Resultado frente a otros enfoques . . . . .	66
4.4. Objetivos Logrados . . . . .	69
4.4.1. Objetivos Específicos . . . . .	69
4.4.2. Objetivo General . . . . .	70
4.5. Problemas encontrados . . . . .	72
<b>5. Conclusiones y trabajos futuros</b>	<b>74</b>
5.1. Conclusiones . . . . .	74
5.2. Impacto social y medioambiental . . . . .	74
5.3. Líneas futuras . . . . .	75
<b>Bibliografía</b>	<b>77</b>

# Índice de tablas

3.1. Comparación de Flair y Sentence Transformers . . . . .	19
3.2. Tripletas con resolvedor de correferencias . . . . .	23
3.3. Tripletas sin resolvedor de correferencias . . . . .	23

# Índice de figuras

1.1. Aplicaciones de los KG [1] . . . . .	2
1.2. Objetivos específicos . . . . .	5
2.1. Tareas principales de IE [2] . . . . .	6
2.2. Ejemplo de extracción de cláusulas mediante el enfoque utilizado en Stanford OpenIE [3].	9
2.3. Línea del tiempo de principales herramientas de OpenIE . . . . .	10
2.4. Ejemplo del análisis realizado por Stanford CoreNLP . . . . .	11
2.5. Proceso simplificado de uso de Neo4j LLM Knowledge Graph Builder [4] . . . . .	13
3.1. Métrica de precisión para grafos . . . . .	21
3.2. Métrica de cobertura para grafos . . . . .	21
3.3. Métrica de similitud estructural para grafos . . . . .	21
3.4. Distribuciones de confianza de las tripletas de los grafos Baseline . . . . .	31
4.1. Evaluación inicial de grafos - Utilitarianism . . . . .	34
4.2. Evaluación inicial de grafos - Ellipse . . . . .	34
4.3. Evaluación inicial de grafos - Homeostasis . . . . .	35
4.4. Evaluación inicial de grafos - Inflation . . . . .	36
4.5. Utilitarianism: Baseline vs Estrategia 1 . . . . .	39
4.6. Ellipse: Baseline vs Estrategia 1 . . . . .	39
4.7. Homeostasis: Baseline vs Estrategia 1 . . . . .	40
4.8. Inflation: Baseline vs Estrategia 1 . . . . .	40
4.9. Evolución media de las métricas con Estrategia 1 . . . . .	41
4.10. Utilitarianism: Baseline vs Estrategia 2 . . . . .	42
4.11. Ellipse: Baseline vs Estrategia 2 . . . . .	43
4.12. Homeostasis: Baseline vs Estrategia 2 . . . . .	43
4.13. Inflation: Baseline vs Estrategia 2 . . . . .	44
4.14. Evolución media de las métricas Estrategia 2 . . . . .	45
4.15. Evaluación Bloque I . . . . .	46
4.16. Utilitarianism: Estrategia 1 vs Estrategia 1 + 3 . . . . .	47
4.17. Ellipse: Estrategia 1 vs Estrategia 1 + 3 . . . . .	48
4.18. Evolución media de las métricas Estrategia 3 . . . . .	49
4.19. Utilitarianism: Estrategia 1 vs Estrategia 1 + 4 . . . . .	50
4.20. Ellipse: Estrategia 1 vs Estrategia 1 + 4 . . . . .	50
4.21. Homeostasis: Estrategia 1 vs Estrategia 1 + 4 . . . . .	51
4.22. Inflation: Estrategia 1 vs Estrategia 1 + 4 . . . . .	51
4.23. Evolución media de las métricas Estrategia 4 . . . . .	52
4.24. Utilitarianism: Estrategia 1 vs Estrategia 1 + 5 . . . . .	53
4.25. Ellipse: Estrategia 1 vs Estrategia 1 + 5 . . . . .	54
4.26. Homeostasis: Estrategia 1 vs Estrategia 1 + 5 . . . . .	54
4.27. Inflation: Estrategia 1 vs Estrategia 1 + 5 . . . . .	55
4.28. Evolución media de las métricas Estrategia 5 . . . . .	56
4.29. Evaluación Bloque II . . . . .	57
4.30. Utilitarianism: Estrategia 1 + 3 + 5 vs Estrategia 1 + 3 + 5 + 6 . . . . .	58
4.31. Ellipse: Estrategia 1 + 3 + 5 vs Estrategia 1 + 3 + 5 + 6 . . . . .	59
4.32. Homeostasis: Estrategia 1 + 3 + 5 vs Estrategia 1 + 3 + 5 + 6 . . . . .	59
4.33. Inflation: Estrategia 1 + 3 + 5 vs Estrategia 1 + 3 + 5 + 6 . . . . .	60

4.34. Evolución media de las métricas Estrategia 6 . . . . .	60
4.35. Utilitarianism: Estrategia 1 + 3 + 5 vs Estrategia 1 + 3 + 5 + 7 . . . . .	61
4.36. Ellipse: Estrategia 1 + 3 + 5 vs Estrategia 1 + 3 + 5 + 7 . . . . .	62
4.37. Homeostasis: Estrategia 1 + 3 + 5 vs Estrategia 1 + 3 + 5 + 7 . . . . .	62
4.38. Inflation: Estrategia 1 + 3 + 5 vs Estrategia 1 + 3 + 5 + 7 . . . . .	63
4.39. Evolución media de las métricas Estrategia 7 . . . . .	64
4.40. Evaluación Bloque III . . . . .	65
4.41. Evaluación final de grafos - Utilitarianism . . . . .	66
4.42. Evaluación final de grafos - Ellipse . . . . .	67
4.43. Evaluación final de grafos - Homeostasis . . . . .	68
4.44. Evaluación final de grafos - Inflation . . . . .	68
4.45. Evolución de las métricas por bloque. . . . .	69
4.46. Evaluación final promedio para el dataset de referencia . . . . .	71
5.1. ODS relacionados con este PFG . . . . .	75

# Capítulo 1

## Introducción

### 1.1. Contexto

Los grafos de conocimiento (KG, por sus siglas en inglés) son representaciones estructuradas que modelan hechos del mundo real en forma de entidades y relaciones. Formalmente, un KG se puede definir como un grafo multirrelacional, donde los nodos representan entidades (como personas, lugares o conceptos) y los arcos etiquetados corresponden a relaciones entre ellas [5].

Gracias a esta estructura, las máquinas pueden “entender” mejor los datos, llegar a nuevas conclusiones y tomar decisiones más informadas. Además, al unir y relacionar gran cantidad de datos procedentes de fuentes muy distintas, los KG facilitan un análisis profundo y un uso más eficaz de la información en proyectos de gran escala, tanto en la industria como en la investigación académica [1].

Un buen ejemplo de lo poderosos que pueden ser estos grafos es el Knowledge Graph de Google. Se trata de una base de datos que Google emplea para mostrar en los resultados de búsqueda información clave sobre personas, lugares, obras, eventos y mucho más. De este modo, el usuario obtiene datos útiles directamente en la propia página de resultados, sin necesidad de hacer clic en otros enlaces. Además, el usuario no tiene que mencionar la entidad de forma explícita. Google, gracias a su grafo y otras técnicas de procesamiento del lenguaje, deduce el significado de la consulta subyacente y ofrece la información más adecuada [6].

Hoy en día, los KG se usan en múltiples aplicaciones prácticas. Por ejemplo, en los sistemas de pregunta-respuesta, el grafo permite extraer respuestas precisas basándose en las relaciones semánticas entre los conceptos. También se utilizan en los motores de recomendación, aprovechándose las conexiones entre usuarios, productos y atributos para sugerir contenidos o servicios relevantes. O en los buscadores de información como Google, que enriquecen los resultados al ir más allá de una simple coincidencia de palabras y tener en cuenta también la relación entre conceptos.

Los KG también tienen aplicaciones en sectores específicos como la medicina, la ciberseguridad, las finanzas, las noticias o la educación. En medicina, integran datos clínicos y terminologías para automatizar la extracción de hechos y mejorar diagnósticos y seguimientos. En ciberseguridad, combinan ontologías de ataques, alertas y vulnerabilidades para detectar patrones y anticipar incidentes. En finanzas, enlazan noticias, entidades empresariales y flujos de mercado para modelar riesgos, identificar fraudes y prever movimientos de acciones. En el ámbito de las noticias, extraen entidades y eventos de múltiples fuentes para seguir la evolución de sucesos y evaluar la veracidad de la información. Por último, en educación, representan conceptos, competencias y materiales de aprendizaje para recomendar recursos personalizados y visualizar rutas formativas.

Estas aplicaciones están impulsando la importancia de generar KG a partir de texto, ya que este enfoque permite convertir el enorme volumen de información no estructurada en información estructurada mediante técnicas de procesamiento de lenguaje natural (NLP, por sus siglas en inglés). Este proceso de construcción de grafos de conocimiento a partir de texto, también conocido como Knowledge Graph Construction (KGC), consiste en identificar entidades, relaciones y eventos relevantes en los textos y representarlos en forma de triplets sujeto-relación-objeto que unidas formen una fuente de conocimiento estructurada. [7].

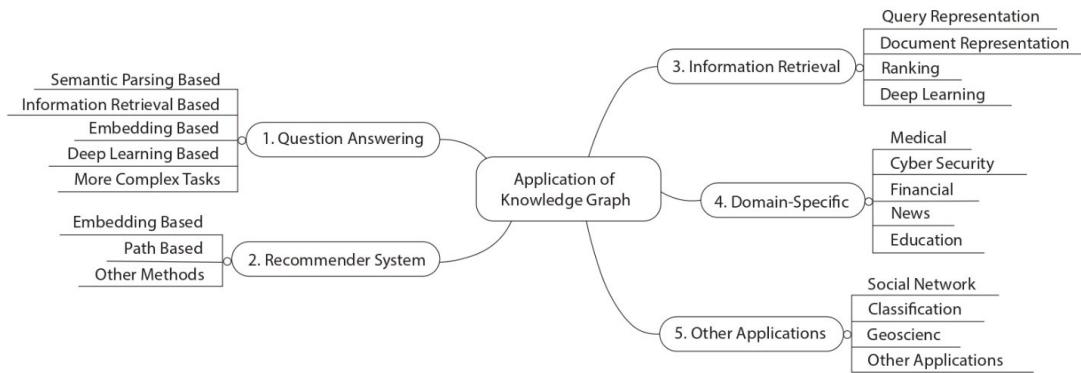


Figura 1.1: Aplicaciones de los KG [1]

En este contexto, una herramienta especialmente relevante ha sido Open Information Extraction (OpenIE), un enfoque que permite extraer relaciones directamente desde el texto sin necesidad de esquemas predefinidos ni dominios específicos. OpenIE ha sido ampliamente adoptado como componente inicial en muchos sistemas de KGC, ya que facilita la extracción masiva de triplets relacionales de forma no supervisada. No obstante, su dependencia de técnicas lingüísticas tradicionales también implica cierta vulnerabilidad ante ambigüedad semántica, errores de análisis sintáctico o falta de contexto [8].

Los grandes modelos de lenguaje (LLMs, por sus siglas en inglés) han ganado popularidad en los últimos años, impulsando aplicaciones capaces de procesar y comprender texto a gran escala. El uso de LLMs para canalizar y estructurar la información es uno de los enfoques más prometedores para la generación de KG a partir de texto. En este sentido, Neo4j LLM Knowledge Graph Builder se presenta como una solución web desarrollada por Neo4j Labs que integra LLMs para transformar automáticamente documentos, páginas web, transcripciones de vídeo y otros formatos textuales en un grafo de conocimiento. La herramienta orquesta llamadas al modelo de lenguaje para identificar entidades, relaciones y atributos, generando las triplets correspondientes y cargándolas en una base de datos Neo4j, lo que permite explorar y consultar la información de manera semántica y visual [9].

Por otro lado, Notebook LM, presentado por Google Labs en 2023, ha ganado popularidad como asistente de investigación basado en LLMs. Aunque no está diseñado específicamente para la construcción de grafos de conocimiento, Notebook LM permite al usuario importar sus propios documentos (por ejemplo, apuntes en Google Docs o PDFs) y, a partir de ellos, generar resúmenes, explicaciones de conceptos complejos y conexiones implícitas entre ideas[10].

Este Proyecto de Fin de Grado (PFG) propone un método ligero para construir grafos de conocimiento a partir de texto sin depender exclusivamente de LLMs. La aproximación se basa en un pipeline que parte de la extracción de triplets mediante OpenIE y aplica un conjunto de estrategias lingüísticas y estructurales con bajo coste computacional, diseñadas para enriquecer y refinar el grafo resultante. Estas estrategias se agrupan en tres bloques: expansión de cobertura, homogeneización y refinamiento, y mejora de precisión. La calidad del grafo generado se evalúa mediante métricas específicas de precisión, cobertura y similitud estructural, aplicadas en el plano léxico y semántico.

## 1.2. Motivación

Generar KG directamente desde texto se ha convertido en un paso importante para sacar partido a la ingente cantidad de información no estructurada que existe. Cada día se publican y archivan millones de documentos como artículos científicos, informes clínicos, noticias, etc. en formato no estructurado. Transformar esta información en conocimiento estructurado mediante grafos de conocimiento permite que conceptos y relaciones queden explícitos, puedan enlazarse entre sí y sean directamente consultables por máquinas. Esto acelera la búsqueda de respuestas precisas y facilita la integración de fuentes heterogéneas, además de habilitar el “razonamiento automático” para detectar patrones que resultarían invisibles en texto plano. Es decir, estructurar la información en forma de grafo desbloquea un enorme potencial que motiva a este PFG.

Sin embargo, los mejores resultados publicados durante los últimos años para KGC dependen de LLMs, cuyo uso plantea varios retos de fondo:

1. **Coste computacional y huella de carbono.** Entrenar y afinar LLMs requiere grandes cantidades de energía y hardware especializado, lo que se traduce en una huella de carbono considerable y un impacto ambiental negativo. [11]
2. **Barreras de acceso y soberanía tecnológica.** El hardware de alto rendimiento es costoso y escaso. Además, muchas APIs de LLMs añaden tarifas por token y restricciones que complican la investigación abierta y la autonomía tecnológica.
3. **Necesidad de alternativas ligeras y explicables.** En entornos con recursos limitados resulta inviable mantener LLMs. Métodos clásicos como OpenIE, que se ejecutan en CPU y procesan texto a gran velocidad, permiten pipelines transparentes, reproducibles y de bajo consumo.

En este PFG se presenta un enfoque práctico y ligero para la generación de grafos de conocimiento a partir de texto, sin recurrir únicamente a grandes modelos de lenguaje. La propuesta parte de la extracción de triplets mediante OpenIE y se basa en un pipeline diseñado para enriquecer y refinar esas triplets mediante una serie de estrategias lingüísticas y estructurales de bajo coste computacional. Estas estrategias se organizan en tres bloques: expansión de cobertura, homogeneización y refinamiento, y mejora de precisión.

Para el diseño de las estrategias que conforman el pipeline de generación de grafos, se parte de una serie de hipótesis sobre cómo cada transformación puede contribuir a mejorar la calidad del grafo resultante. Estas estrategias se implementan y evalúan de forma progresiva mediante métricas específicas de precisión, cobertura y similitud estructural, tanto en el plano léxico como en el semántico.

## 1.3. Objetivos

### 1.3.1. Objetivo general

El objetivo principal de este proyecto es diseñar un pipeline ligero para la generación de grafos de conocimiento a partir de texto, basado en OpenIE y sin recurrir al uso de grandes modelos de lenguaje (O1). El sistema propuesto parte de la extracción de tripletas sujeto-predicado-objeto mediante OpenIE, que son posteriormente enriquecidas mediante un conjunto de estrategias lingüísticas y estructurales de bajo coste computacional. La calidad del grafo resultante se evaluará mediante métricas específicas de precisión, cobertura y similitud estructural, comparándolo con un grafo de referencia (Gold) elaborado manualmente, con el fin de validar la efectividad del enfoque propuesto.

### 1.3.2. Objetivos específicos

Para desarrollar y validar el pipeline ligero propuesto para la generación de grafos de conocimiento, se plantean los siguientes objetivos concretos:

1. **Incrementar la cobertura del grafo (O1.1).** Ampliar la representación del conocimiento identificando tripletas que OpenIE pasa por alto en su primera extracción sobre el dataset de referencia. Esto implica aplicar técnicas de segmentación sintáctica, re-extracción y detección de verbos para capturar relaciones adicionales. El éxito se medirá por el aumento de la cobertura semántica y léxica en el grafo Baseline frente al grafo Gold.
2. **Refinar y homogeneizar relaciones y entidades (O1.2).** Corregir predicados mal formados y fusionar menciones equivalentes (incluyendo variaciones morfológicas) para reducir la ambigüedad y mejorar la cohesión del grafo. Para ello, se aplicarán reglas lingüísticas y un mapeo de entidades. La mejora se evaluará mediante las métricas de precisión, cobertura y similitud estructural respecto al grafo Gold.
3. **Mejorar la precisión del grafo (O1.3).** Eliminar tripletas redundantes o de baja confianza, conservando solo las relaciones con mayor evidencia textual. El objetivo es incrementar la precisión léxica y semántica y reducir el ruido estructural. La mejora se comprobará a través de las métricas de precisión (léxica y semántica) respecto al grafo Gold.
4. **Evaluar el impacto de las estrategias de mejora (O1.4).**
  - a) Medir, para cada estrategia del pipeline, la variación en precisión, cobertura y similitud estructural respecto al grafo Baseline y al grafo Gold.
  - b) Comparar el pipeline completo con los grafos generados por Neo4j LLM Graph Builder y NotebookLM, cuantificando la ganancia obtenida.

## 1.4. Estructura del documento

Este documento se estructura en seis capítulos, cada uno orientado a desarrollar de forma progresiva los distintos aspectos del trabajo realizado. A continuación se detalla el contenido de cada uno de ellos:

- **Capítulo 1 – Introducción:** Presenta el contexto en el que se enmarca el trabajo, la motivación que lo impulsa, así como los objetivos generales y específicos planteados.
- **Capítulo 2 – Estado del arte:** Revisa los enfoques existentes para la generación de grafos de conocimiento a partir de texto, con especial atención a los métodos basados en OpenIE y los sistemas apoyados en modelos de lenguaje de gran escala (LLMs). Se analizan también técnicas relacionadas con las estrategias propuestas en este proyecto.

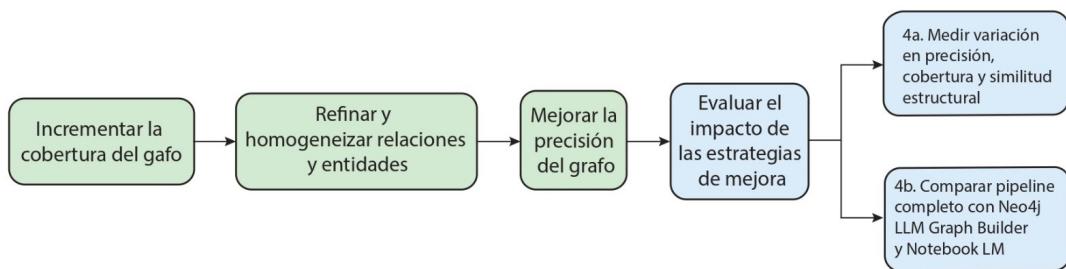


Figura 1.2: Objetivos específicos

- **Capítulo 3 – Metodología:** Describe en detalle el diseño experimental seguido. Se introduce el dataset de referencia utilizado para la evaluación, las métricas de comparación aplicadas y los distintos tipos de grafos generados: el grafo Gold, el Baseline obtenido mediante OpenIE, y dos aproximaciones generadas con herramientas basadas en LLMs. Finalmente, se explican las estrategias propuestas para mejorar el grafo Baseline, agrupadas en tres bloques.
- **Capítulo 4 – Resultados:** Presenta los resultados obtenidos en las distintas fases del proyecto. Se analiza en primer lugar el rendimiento del Baseline frente a los sistemas basados en LLMs. A continuación, se evalúa el impacto de cada bloque de estrategias sobre las métricas consideradas, y se comparan los resultados finales. También se reflexiona sobre el cumplimiento de los objetivos y se describen los principales problemas encontrados.
- **Capítulo 5 – Conclusiones y trabajos futuros:** Resume las principales aportaciones del proyecto y ofrece una valoración crítica de los resultados. Además, se discute el impacto potencial de la propuesta y se proponen posibles líneas de mejora y ampliación para trabajos futuros.
- **Capítulo 6 – Bibliografía:** Recoge las fuentes bibliográficas y trabajos previos que han servido de base para el desarrollo de este proyecto.

# Capítulo 2

## Estado del arte

### 2.1. Construcción de KGs a partir de texto: Visión general

La construcción de Grafos de Conocimiento (KGC) es un área de investigación activa que busca representar la información contenida en grandes corpus textuales de manera más organizada, estructurada y compacta. El propósito de esta representación es facilitar su uso eficiente en diversas aplicaciones, como el razonamiento automático, la toma de decisiones y la respuesta a preguntas. Construir un KG a partir de texto implica transformar información no estructurada en información estructurada, típicamente en forma de triplets del tipo sujeto–relación–objeto.

Para lograr la construcción de un grafo, tradicionalmente el problema se descompone en varias tareas, entre las cuales destacan: el reconocimiento de entidades nombradas (Named Entity Recognition, NER), la extracción de relaciones (Relation Extraction, RE), la extracción de eventos (EE), el enlace de entidades a identificadores únicos en una base de conocimiento (Entity Linking), y la completitud o enriquecimiento del grafo, añadiendo relaciones o nodos faltantes [12].

Las tres primeras tareas forman el núcleo de la construcción del grafo, basado en la extracción de information (IE, Information Extraction) ya que permiten definir los nodos (entidades) y los arcos (relaciones y eventos) [2].

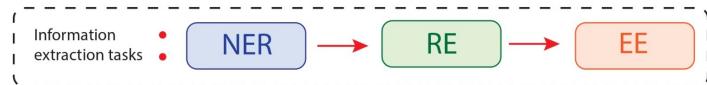


Figura 2.1: Tareas principales de IE [2]

Las tareas principales para la extracción de información son:

- **Reconocimiento de Entidades Nombradas (NER).** Comprende dos tareas: identificación de entidades (detectar los fragmentos de texto que corresponden a entidades) y clasificación de entidades (asignar un tipo a cada entidad identificada).
- **Extracción de Relaciones (RE).** Puede abordarse de diversas formas:
  1. Clasificación de relaciones: clasificar el tipo de relación entre dos entidades dadas.
  2. Tripletas de relación: identificar el tipo de relación junto con los fragmentos (*spans*) de las entidades origen (*head*) y destino (*tail*).
  3. Relación estricta: además de lo anterior, también identificar el tipo de las entidades implicadas.
- **Extracción de Eventos (EE):** consiste en identificar los eventos que ocurren en el texto y clasificar qué tipo de eventos son. Una parte clave de esta tarea es detectar la palabra que indica que un evento ocurre (llamada *trigger*).

Estos métodos (NER, RE y EE) han sido tradicionalmente implementados mediante clasificadores entrenados y sistemas basados en reglas, y constituyen la base del enfoque convencional de KGC.

Sin embargo, este enfoque por etapas presenta limitaciones, como la propagación de errores entre tareas y la dificultad de mantener coherencia estructural a gran escala.

Existen principalmente dos paradigmas para la construcción de grafos de conocimiento a partir de texto. El primero es la extracción supervisada, donde se definen previamente los tipos de relaciones de interés y se entrena nodelos o diseñan reglas específicas para detectarlas en el texto. Aunque este enfoque puede alcanzar alta precisión en dominios específicos, requiere esquemas predefinidos y conjuntos de datos anotados para el entrenamiento, lo que limita su escalabilidad.

El segundo paradigma es la extracción abierta de información (Open Information Extraction, OpenIE), que busca extraer relaciones entre entidades directamente del texto, sin requerir conocimiento previo del dominio, y representarlas como triplets sujeto-predicado-objeto [8]. Los sistemas de OpenIE se basan en la anotación de oraciones y la extracción de triplets a partir de ellas. Estos sistemas, basados principalmente en reglas semánticas, muestran ciertas limitaciones como la creación de triplets redundantes o los errores en los límites de las extracciones.

Para superar estas limitaciones, han surgido enfoques generativos que reformulan la construcción del grafo como un problema de generación de secuencias. En este paradigma, se representa el grafo de forma lineal y se utilizan modelos de lenguaje preentrenados (como T5 o BART) para traducir directamente el texto en secuencias de triplets. Este enfoque basado en modelos *sequence-to-sequence* ha demostrado resultados prometedores para grafos de tamaño pequeño o medio. No obstante, enfrenta retos como la no unicidad en la representación lineal, la redundancia y la complejidad creciente de las secuencias generadas [13].

También se han propuesto métodos alternativos que modelan explícitamente la generación de nodos y relaciones. Un ejemplo es *GraphRNN*, que genera grafos añadiendo secuencialmente nodos y arcos mediante redes neuronales recurrentes (RNNs). Sin embargo, su naturaleza secuencial puede conducir a estructuras subóptimas [14]. Otros enfoques más recientes como CycleGT o DualTKB permiten una generación bidireccional entre texto y grafo, aunque su uso se limita generalmente a oraciones simples que contienen una única tripla.

En los últimos años, los LLMs han demostrado capacidades destacables en la comprensión y generación de texto, lo que ha motivado numerosos trabajos que los aplican tanto a la construcción de grafos como a tareas de extracción de información bajo un paradigma generativo [2].

Estos métodos de KGC a partir de texto han avanzado considerablemente, pero el producto resultante no está exento de errores, omisiones o inconsistencias. Actualmente, no existe un conjunto universalmente aceptado de métricas estandarizadas para evaluar de forma sistemática la calidad de estos grafos generados automáticamente, lo que complica la comparación entre enfoques y la validación de su utilidad práctica.

El estado del arte en la evaluación de grafos de conocimiento generados automáticamente a partir de texto se basa en un conjunto bien definido de dimensiones de calidad, entre las que destacan la precisión, la consistencia, la completitud, la actualidad y la redundancia [15].

Estas dimensiones permiten evaluar desde múltiples ángulos la fidelidad, coherencia y utilidad del grafo respecto al conocimiento que representa. La precisión se asocia al grado de veracidad de los hechos extraídos; la consistencia, a la ausencia de contradicciones internas; la completitud, a la cobertura informativa del dominio objetivo; la actualidad, a la vigencia temporal de los datos; y la redundancia, a la evitación de información duplicada. Dado que estas dimensiones son conceptos abstractos, su implementación requiere definir métricas operativas específicas, como el porcentaje de triplets correctas o la proporción de entidades o relaciones faltantes. Además, estas dimensiones suelen estar interrelacionadas, por lo que optimizar una métrica puede afectar negativamente a otra, lo que plantea desafíos metodológicos en la evaluación de la calidad de los KGs construidos desde texto.

## 2.2. Extracción de Información Abierta

Tradicionalmente, la tarea de extracción de relaciones a partir de texto se ha abordado mediante dos enfoques principales: los sistemas basados en reglas y los métodos de Extracción de Información Abierta (OpenIE). Los métodos basados en reglas dependen de patrones lingüísticos explícitos definidos manualmente, como expresiones regulares, estructuras sintácticas y ontologías. Estos sistemas son altamente precisos en dominios específicos y permiten controlar el tipo de relaciones que se extraen, pero su escalabilidad es limitada y requieren gran esfuerzo de diseño por expertos lingüísticos.

Por su parte, OpenIE representa un enfoque más flexible y escalable, ya que no necesita una lista predefinida de relaciones ni un dominio específico. Desde su aparición en 2007, OpenIE ha evolucionado desde sistemas que usaban herramientas lingüísticas básicas hasta métodos que incorporan características sintácticas y semánticas complejas.

### 2.2.1. Desde TextRunner hasta los LLMs

En 2007 nace OpenIE con la primera generación representada por **TextRunner** [16]. Este sistema buscaba extraer información estructurada en forma de triplets sujeto-relación-objeto a partir de texto no estructurado, sin restricciones en cuanto al tipo de relación o dominio. El funcionamiento de TextRunner se basaba en los siguientes pasos:

- **Single Pass Extractor (Extracción en una sola pasada):**

TextRunner procesaba los documentos en una única pasada, aplicando técnicas de análisis superficial como el etiquetado gramatical (POS tagging) y la segmentación de frases nominales (noun phrase chunking) a cada oración.

- **Self-Supervised Classifier (Clasificación auto-supervisada):**

Utilizaba un analizador sintáctico a menor escala para generar automáticamente sus propios ejemplos de entrenamiento, distinguiendo entre extracciones confiables y no confiables mediante restricciones heurísticas. Posteriormente, entrenaba un clasificador con estos ejemplos, empleando características como las etiquetas POS para evaluar si una secuencia de palabras representaba una relación válida entre frases nominales. Si el clasificador determinaba que una relación era confiable, el sistema extraía una triplete con la estructura sujeto-relación-objeto.

- **Synonym Resolution (Resolución de sinónimos):**

Debido a la ausencia de un conjunto predefinido de relaciones, el sistema podía extraer múltiples variantes para la misma relación u objeto. Para resolver esto, se incluía un componente llamado *Resolver*, encargado de agrupar automáticamente las extracciones mediante técnicas no supervisadas, identificando sinónimos o expresiones equivalentes.

Aunque fue un gran avance en su momento, los primeros sistemas de Open IE como TextRunner solían generar extracciones incoherentes o poco informativas. Estas limitaciones motivaron el desarrollo de sistemas posteriores, como ReVerb, que incorporaron un análisis lingüístico más profundo y restricciones más precisas para mejorar la calidad de las extracciones.

Alrededor de 2011, después de casi cinco años, surge la segunda generación de sistemas OpenIE, con sistemas como ReVerb [17]. Se enfocan en mejorar la coherencia y la informatividad de las extracciones mediante un diseño más guiado lingüísticamente y el uso de características lingüísticas más profundas. ReVerb, como sistema de segunda generación de Open IE, fue diseñado a partir de un análisis lingüístico exhaustivo sobre cómo se expresan las relaciones en inglés.

Su funcionamiento incorporó varias innovaciones clave:

- **Restricciones sintácticas:** ReVerb utiliza una novedosa restricción sintáctica basada en un patrón específico de etiquetas gramaticales (Part-of-Speech), que exige que las frases de relación extraídas sean una secuencia coherente y continua de palabras. Esto permite eliminar extracciones incoherentes.
- **Restricciones léxicas:** Además de analizar la estructura gramatical, ReVerb aplica una restricción léxica que se asegura de que las frases de relación extraídas no sean demasiado específicas. Para ello, solo conserva aquellas que aparecen con muchos sujetos y objetos distintos en un gran corpus de texto. Esto permite que las relaciones extraídas sean más generales, reutilizables y útiles en distintos contextos.
- **Relation First Approach:** A diferencia de TextRunner, que identificaba primero los argumentos y luego clasificaba las palabras entre ellos, ReVerb adopta un enfoque de *relation first*. Es decir, identifica primero las frases de relación que cumplen con sus restricciones, tratándolas como una unidad completa, y luego busca las frases nominales más cercanas que actúen como argumentos.

En 2012 se publica el artículo *Open Language Learning for Information Extraction* [18], en el que se presenta OLLIE, un sistema diseñado para superar dos limitaciones clave de sistemas anteriores. Por un lado, OLLIE amplía el alcance sintáctico de la extracción, permitiendo identificar relaciones no solo mediadas por verbos, sino también por sustantivos, adjetivos y otros elementos gramaticales, lo que incrementa considerablemente la cantidad de triplets extraídas. Por otro lado, introduce un análisis contextual que permite capturar información adicional, como atribuciones (*attributions*) y modificadores oracionales (*clausal modifiers*), mejorando así la precisión y la veracidad de las extracciones.

En 2013, se propone ClausIE (*Clause-Based Open Information Extraction*) [19], que introduce un enfoque fundamentalmente distinto: separa la detección de información útil en una oración de su representación como tripleta. ClausIE se basa en el conocimiento lingüístico de la gramática del inglés para, primero, detectar las cláusulas dentro de una oración y, luego, clasificar su tipo según las funciones gramaticales de sus componentes (como sujeto, verbo y objeto).

Posteriormente, en 2015, el artículo *Leveraging Linguistic Structure for Open Domain Information Extraction* [3] sienta las bases del módulo OpenIE de Stanford CoreNLP (SIE). Aprovecha la estructura lingüística mediante la división de oraciones en cláusulas coherentes y el uso de la lógica natural para refinar las extracciones.

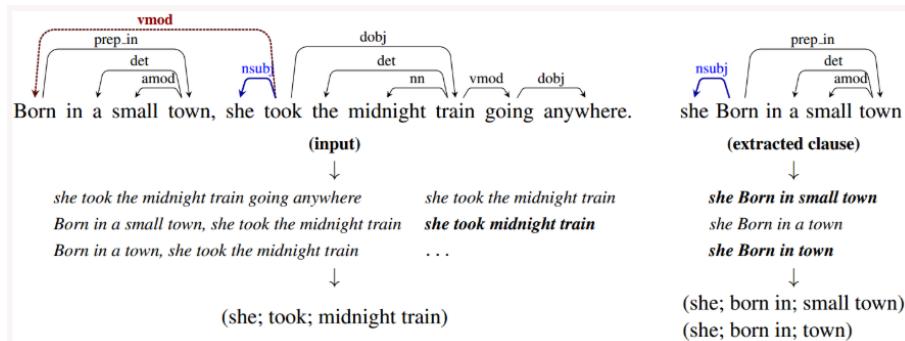


Figura 2.2: Ejemplo de extracción de cláusulas mediante el enfoque utilizado en Stanford OpenIE [3].

Una limitación conocida del módulo Stanford OpenIE es su tendencia a generar extracciones redundantes a partir de una misma oración. Esto se debe a que está optimizado para lograr un alto nivel de cobertura.

Esto implica que, para no dejar pasar posibles relaciones, el sistema puede generar muchas tripletas para una misma estructura sujeto-predicado. Si bien esto aumenta la cobertura, también puede incrementar la repetición de resultados.

A pesar de ello, Stanford OpenIE puede considerarse uno de los sistemas más representativos del estado del arte entre los enfoques de OpenIE que no emplean modelos de lenguaje de gran escala, destacando por su eficiencia, integración práctica y aprovechamiento del análisis lingüístico tradicional. Esto se debe a que el auge de los modelos neuronales en 2019 marcó un cambio significativo, integrando arquitecturas como BERT para mejorar la calidad de las tripletas extraídas [20].

Más recientemente, la aparición de modelos de lenguaje de gran escala LLMs y su popularización en 2023 han llevado a una transformación del campo. Estos modelos generativos permiten realizar extracción relacional directamente desde el texto, acercándose a un enfoque unificado y multitarea.

Gracias a su alta capacidad de generalización, los LLMs aportan avances significativos en tareas fundamentales para la construcción de grafos de conocimiento (KGs), como el NER, RE y EE [21]. Además, los LLMs abren nuevas oportunidades de futuro, como el desarrollo de AutoKG [22], un concepto emergente que plantea la construcción y el razonamiento autónomos de grafos de conocimiento mediante comunicación entre múltiples agentes inteligentes.

A pesar de ello, los enfoques clásicos de OpenIE, especialmente los basados en reglas o estructuras sintácticas, siguen siendo valiosos como alternativa ligera y comprensible, y pueden inspirar desarrollos futuros más eficientes y controlables frente al uso intensivo de LLMs [8].

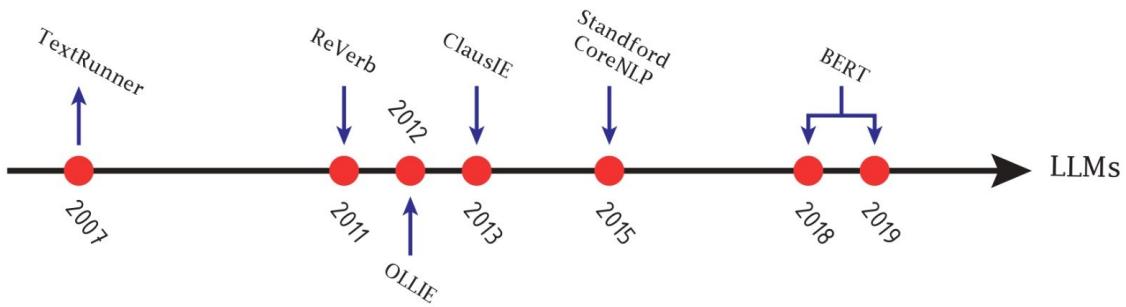


Figura 2.3: Línea del tiempo de principales herramientas de OpenIE

### 2.2.2. Proceso Básico OpenIE

El proceso central de un sistema OpenIE generalmente consiste en un pipeline que procesa texto en lenguaje natural a través de dos pasos principales: la anotación de oraciones y la creación de tripletas [23].

La anotación de oraciones se puede hacer de dos maneras:

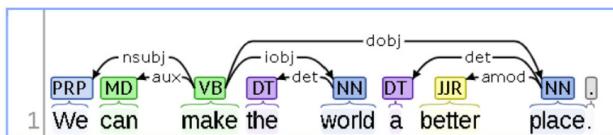
- **Análisis Superficial (Shallow Analysis):** Utiliza algoritmos que identifican partes de la oración, como el etiquetado de categorías gramaticales (POS tagging) y la segmentación de frases nominales o verbales. Este enfoque trata la oración como un árbol, pero permite el acceso solo a partes específicas de forma aislada.

- **Análisis de Dependencias (Dependency Analysis):** Establece relaciones entre todas las palabras de una oración, partiendo de una raíz (generalmente un verbo) desde la cual se puede acceder a todos los elementos. Aunque es más costoso computacionalmente que el análisis superficial, los sistemas que utilizan este enfoque, como ClausIE y OLLIE, han demostrado mejores resultados.

### Part-of-Speech:

	PRP	MD	VB	DT	NN	DT	JJR	NN	
1	We	can	make	the	world	a	better	place.	

### Basic Dependencies:



Tags used in the example	
Part-of-Speech	Basic Dependencies
PRP personal pronoun	aux auxiliary
MD modal	nsubj nominal subject
VB verb, base form	iobj indirect object
DT determiner	dobj direct object
NN noun, singular or mass	det determiner
JJR adjective comparative	amod adjectival modifier
.	

Figura 2.4: Ejemplo del análisis realizado por Stanford CoreNLP

Una vez analizada la oración, los sistemas emplean distintos métodos para extraer las tripletas sujeto-relación-objeto:

- **Métodos de Aprendizaje Automático:** El primer sistema Open IE, TextRunner, utilizaba un modelo de *Naive Bayes* entrenado con ejemplos generados heurísticamente a partir de un conjunto de datos.
- **Reglas Manuales:** A partir del sistema ReVerb de 2011, se empezó a promover el uso de reglas manuales, comúnmente implementadas mediante expresiones regulares, lo que puede mejorar la precisión.
- **Modelos de lenguaje:** Recientemente, la extracción de tripletas ha pasado a ser una tarea realizada por los modelos de lenguaje, que permiten llevar a cabo la extracción relacional directamente desde el texto, sin necesidad de recurrir a la fase tradicional de anotación manual de oraciones.

### 2.2.3. Desafíos de OpenIE

Uno de los principales retos está relacionado con la evaluación y el diseño de estos sistemas. Elegir la herramienta adecuada puede ser complejo, ya que las evaluaciones comparativas han demostrado que ningún sistema OpenIE supera de forma consistente a los demás en todos los conjuntos de datos. Cada uno tiende a destacar en contextos específicos, lo que dificulta su selección para aplicaciones generales. A esto se suma la falta de políticas de anotación rigurosas, ya que las diferencias en los estilos de anotación entre sistemas y estándares de referencia (gold standards) generan inconsistencias, errores y obstáculos en el desarrollo de sistemas más precisos. Además, como ya se ha visto, los sistemas OpenIE suelen depender de herramientas intermedias, como analizadores de dependencias o etiquetadores gramaticales (POS taggers), cuyos errores se propagan y afectan directamente la calidad de las tripletas extraídas.

En cuanto al rendimiento, los sistemas OpenIE presentan varias clases comunes de errores. Uno de los más frecuentes son los límites incorrectos en las extracciones. Estos ocurren cuando los argumentos o predicados tienen un alcance demasiado amplio o estrecho, ya sea por errores en las estructuras intermedias o por una sobreestimación durante la generación de candidatos. Este tipo de error complica tareas posteriores como la fusión, vinculación o agregación de triplets, lo que a menudo obliga a aplicar filtros que, aunque mejoren la precisión de las triplets, reducen considerablemente la cobertura.

Otro problema habitual es la extracción redundante. Los sistemas generan múltiples versiones muy similares de una misma relación dentro de una sola oración. En casos extremos, se han registrado sistemas que producen hasta 140 triplets por oración, en parte porque están diseñados para maximizar la cobertura, dejando la eliminación de redundancias para etapas posteriores del procesamiento. Por último, también se observa el fenómeno de la extracción poco informativa, en el que las triplets generadas omiten información clave. Esto suele suceder en construcciones como “primera (guerra mundial)” o en frases que combinan verbos con sustantivos, como aquellas que utilizan verbos de apoyo. Estos errores reflejan una deficiente interpretación de las estructuras relacionales, lo que limita la utilidad de las extracciones en tareas más avanzadas de procesamiento del lenguaje [24].

### 2.3. Enfoques basados en LLMs

La tarea de KGC a partir de texto resulta especialmente compleja debido al gran volumen y diversidad de la información textual. La aparición de los LLMs, como GPT-3, ChatGPT y GPT-4, ha impulsado significativamente el PLN, permitiendo trabajar con grandes cantidades de texto no estructurado. Esto ha fomentado su uso creciente en la automatización del proceso de KGC [25].

Sin embargo, los primeros enfoques basados en LLMs presentaron limitaciones importantes. Frecuentemente carecían de conocimiento factual preciso, lo que derivaba en “alucinaciones”, y no lograban capturar adecuadamente las complejidades estructurales de los grafos. Además, dependían de esquemas predefinidos y se restringían al procesamiento de un solo documento. Contaban con acceso limitado a conocimiento externo, lo cual reducía su eficacia en dominios especializados. Actualmente, el estado del arte en KGC con LLMs muestra avances significativos mediante soluciones innovadoras que abordan estos retos. Entre ellas destaca la ingeniería de prompts o *prompt engineering*, que es clave para guiar el comportamiento de los modelos y mejorar la calidad de las triplets generadas.

En los desarrollos más recientes, los LLMs no se utilizan necesariamente para realizar toda la tarea de construcción del grafo, sino que apoyan tareas específicas como la predicción de triplets faltantes o la validación de triplets generadas por otros sistemas.

Entre los avances más destacados se encuentran herramientas como TEXT2KGBENCH, un benchmark diseñado para evaluar la generación de grafos de conocimiento a partir de texto, guiado por ontologías [26]. Otro ejemplo es LKD-KGC, un enfoque que analiza repositorios documentales para inferir dependencias de conocimiento. Determina secuencias óptimas de procesamiento mediante la priorización asistida por LLMs y genera esquemas de entidades de manera autorregresiva, sin depender de esquemas manuales o conocimiento externo [27]. Por su parte, *AutoKG* propone un sistema autónomo de construcción de KGs basado en una colaboración multiagente, donde cada agente cumple un rol específico y se comunica con los demás para llevar a cabo la tarea de forma eficiente. Este modelo también permite integrar fuentes externas, como bases de conocimiento o motores de búsqueda en línea, con el fin de superar el corte de conocimiento de los LLMs, mejorar la precisión factual en dominios específicos y aumentar la transparencia del proceso. A pesar de estos avances, los resultados generados aún requieren validación manual para garantizar su fiabilidad [22].

Más recientemente, se ha presentado también *Neo4j LLM Knowledge Graph Builder*, una aplicación web desarrollada por Neo4j Labs que emplea modelos de lenguaje como OpenAI, Gemini, Claude, Llama3, Qwen o Diffbot para procesar documentos, páginas web o transcripciones de vídeo, y convertirlos en grafos de conocimiento. Esta herramienta no solo permite construir el grafo, sino también interactuar con los datos extraídos mediante técnicas de Generación Aumentada por Recuperación (RAG), como *GraphRAG*, *Vector* y *Text2Cypher*, lo que ofrece una experiencia integral de exploración semántica y consulta en lenguaje natural. Su arquitectura combina una interfaz frontend en React con un backend en Python FastAPI desplegado en Google Cloud Run, y utiliza internamente el módulo *llm-graph-transformer*, desarrollado por Neo4j y aportado al ecosistema LangChain, para transformar texto en grafos semánticos [4].

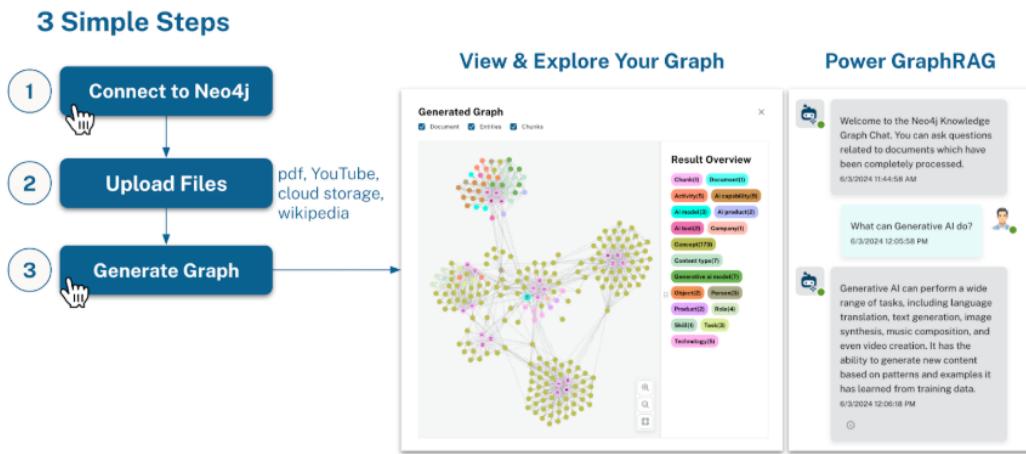


Figura 2.5: Proceso simplificado de uso de Neo4j LLM Knowledge Graph Builder [4]

A pesar de los avances recientes, persisten desafíos al aplicar LLMs en la tarea de KGC. Uno de los principales es la fuerte dependencia de la ingeniería de prompts para obtener resultados relevantes y precisos. Además, el corte de conocimiento inherente a estos modelos limita su acceso a información actualizada o externa al momento de la generación. Otro problema frecuente es el riesgo de producir contenido no factual, las alucinaciones, que afectan a la fiabilidad del grafo generado. Además, las restricciones en la longitud de contexto, como las impuestas por APIs, dificultan el procesamiento de grandes volúmenes de texto de manera continua. Finalmente, muchas de las métricas y conjuntos de datos utilizados para evaluar estos sistemas presentan deficiencias, ya que no logran capturar con precisión todas las capacidades o limitaciones de los LLMs en tareas complejas como el KGC a partir de texto.

## 2.4. Mejoras en la Calidad del Grafo de Conocimiento

Los sistemas de OpenIE han evolucionado significativamente desde su aparición en 2007. Sin embargo, siguen presentando limitaciones reconocidas en cuanto a cobertura semántica, precisión y cohesión estructural, especialmente al enfrentarse a textos complejos o con relaciones implícitas. Se ha observado que la limpieza y el refinamiento de las tripletas extraídas permiten generar grafos de conocimiento más fieles y representativos del contenido textual.

Para abordar estas carencias, se han propuesto distintas técnicas orientadas a mejorar la calidad del grafo generado a partir de las tripletas extraídas. Aunque algunos enfoques recientes se apoyan en grandes modelos de lenguaje, también existen propuestas más ligeras que emplean herramientas clásicas de procesamiento del lenguaje natural, como CoreNLP, SpaCy o reglas heurísticas, lo que las hace especialmente adecuadas para pipelines eficientes y de bajo coste computacional, como el desarrollado en este proyecto.

En este apartado se recopilan y analizan distintas estrategias que permiten ampliar la cobertura del grafo, garantizar una mayor coherencia entre sus componentes y reducir la variabilidad léxica y semántica de las entidades y relaciones.

### Análisis de dependencias para la segmentación de entidades

Este método, utiliza el árbol de análisis sintáctico para descomponer frases nominales complejas en entidades más simples. Este enfoque genera información más detallada y ayuda a reducir la redundancia al crear nodos separados pero conectados para las entidades principales y sus atributos [28].

### Resolución de Correferencias

Esta técnica tiene como objetivo identificar y reemplazar construcciones dependientes del contexto, como pronombres o menciones abreviadas, por la entidad completa a la que hacen referencia. Esto garantiza que todas las referencias a una misma entidad estén vinculadas, proporcionando una representación más completa de la información sobre dicha entidad a lo largo del texto. Para la resolución de correferencias se pueden utilizar métodos basados en reglas, estadísticos y de aprendizaje automático.

### Reconocimiento de Entidades Nombradas (NER)

Identificar entidades nombradas, objetos específicos como personas, organizaciones o eventos, es importante para poder representar información concreta y específica de un dominio. El NER es un paso esencial en los procesos de KGC. Una identificación más exhaustiva de entidades nombradas puede mejorar la etapa de extracción.

### Etiquetado de Roles Semánticos (SRL) [29]

Integrar SRL puede facilitar la extracción de relaciones semánticas al identificar los roles lógicos (por ejemplo, el agente y el paciente) de las entidades dentro de una oración. Esto permite una extracción más estructurada y completa de la información, especialmente al definir quién realiza una acción y quién la recibe.

### Identificación de Sinónimos

Aunque en muchos casos se utiliza BERT para esta tarea, el objetivo principal es identificar sinónimos entre entidades para conectar tripletas que inicialmente están separadas. Otras fuentes sugieren el uso

de recursos externos como WordNet para detectar relaciones de sinonimia e hiperonimia y así agrupar menciones textuales [30].

### Vinculación y Alineación de Entidades

Esto implica enlazar diferentes menciones textuales de una misma entidad del mundo real a un identificador único dentro de una base de conocimiento. La alineación de entidades, conecta entidades de grafos separados que comparten el mismo significado. Para vincular grafos, se puede usar el *Neighborhood matching*, que compara las entidades y relaciones que rodean a cada entidad.

### Integración basada en Sintagmas Nominales (NP)

Esta estrategia consiste en asociar las entidades nombradas con sus unidades gramaticales, en particular los sintagmas nominales (NP), para mantener la cohesión semántica. Esto ayuda a agrupar términos relacionados y asociarlos con el mismo recurso, para asegurar una representación más coherente dentro del grafo.

### Vinculación a Bases de Conocimiento u Ontologías Específicas de Dominio

En dominios especializados, las entidades o relaciones extraídas del texto pueden ser anotadas con identificadores de bases de conocimiento específicas del área. Esto permite resolver ambigüedades y estandarizar los conceptos dentro del dominio.

### Filtrado de Triplets

Con el objetivo de eliminar palabras redundantes o poco informativas dentro de los argumentos, se puede aplicar un filtrado de triplets. Por ejemplo, en el enfoque de OIE4KGC para KGC [29], se construye un léxico con los sustantivos que aparecen con mayor frecuencia en el corpus y se conservan únicamente esas palabras. Este método garantiza que la información restante sea más informativa y concisa, lo que contribuye a un KG más denso y menos disperso.

## Síntesis del Estado del Arte

El estado del arte en la construcción de grafos de conocimiento (KGs) a partir de texto ha evolucionado desde enfoques clásicos basados en reglas y sistemas de extracción abierta de información (OpenIE), como TextRunner o ReVerb, hasta métodos más avanzados impulsados por modelos de lenguaje de gran escala (LLMs). Inicialmente centrados en tareas como NER, RE y EE, los sistemas tradicionales enfrentaban limitaciones en coherencia, cobertura y escalabilidad. Con la llegada de los LLMs, se han desarrollado enfoques generativos, multitarea y autónomos (como AutoKG o Neo4j LLM KG Builder), que mejoran la extracción, generación y razonamiento sobre KGs, aunque aún presentan desafíos como la alucinación, la dependencia del *prompting* y la falta de métricas estandarizadas. Para mejorar la calidad de los grafos generados, se han propuesto técnicas complementarias como el desempaquetado de entidades, la resolución de correferencias, la vinculación a ontologías y el filtrado de triplets. Esto permite generar representaciones más precisas, coherentes y útiles del conocimiento contenido en el texto.

# Capítulo 3

## Metodología

### 3.1. Dataset de Referencia

#### 3.1.1. Objetivo del Dataset de Referencia

El propósito principal de este dataset de referencia es proporcionar un conjunto de textos de diversos temas sobre el cual evaluar y comparar distintos métodos de generación de grafos de conocimiento. Este dataset se emplea para analizar el comportamiento de cuatro grafos construidos a partir de las mismas fuentes textuales, con el fin de cuantificar sus fortalezas y debilidades relativas. Además, se usará el mismo dataset para evaluar, mediante una serie de métricas léxicas y semánticas, las transformaciones que se aplicarán al Baseline con el objetivo de mejorarlo.

El objetivo principal de esta comparación es evaluar en qué medida la herramienta desarrollada con recursos limitados, puede generar grafos de conocimiento que se aproximen tanto a los construidos manualmente por un ser humano como a los generados por sistemas basados en LLMs. De este modo, se busca validar que una solución más ligera puede ofrecer resultados estructurados y coherentes que no se alejen significativamente de la percepción humana ni del rendimiento de modelos de última generación.

Los cuatro grafos contemplados son:

- **Grafo Gold:** construido manualmente, sirve como patrón de calidad y coherencia semántica.
- **Grafo OpenIE (línea base):** extraído automáticamente mediante OpenIE para establecer un punto de partida sin grandes recursos.
- **Grafo Neo4j LLM Graph Builder:** generado a partir de un pipeline asistido por modelos de lenguaje de gran escala, permite valorar el beneficio de los LLMs frente a soluciones ligeras.
- **Grafo Notebook LM:** obtenido mediante otro enfoque basado en LLMs, utilizado como comparación adicional para validar la consistencia de los resultados.

Gracias a este esquema de cuatro grafos, el estudio mide cuantitativamente cómo cada enfoque influye en la precisión, la cobertura y la similitud estructural en los niveles léxico y semántico, tomando el grafo Gold como referencia de calidad.

#### 3.1.2. Descripción del Dataset de Referencia

El dataset de referencia consta de cuatro fragmentos extraídos de textos académicos de distintas disciplinas, seleccionados para garantizar diversidad temática y comparabilidad estructural.

##### 3.1.2.1. Texto 1: Utilitarianism

El texto pertenece a la disciplina de Filosofía moral y procede de la Stanford Encyclopedia of Philosophy [31]. Ofrece una introducción al utilitarismo, una teoría ética normativa que valora las acciones según sus consecuencias y propone maximizar el bien colectivo.

*"Utilitarianism is one of the most powerful and persuasive approaches to normative ethics in the history of philosophy. Though not fully articulated until the 19th century, proto-utilitarian positions can be discerned throughout the history of ethical theory. Though there are many varieties of the view discussed, utilitarianism is generally held to be the view that the morally right action is the action that produces the most good. One thing to note is that the theory is a form of consequentialism: the right action is understood entirely in terms of consequences produced. What distinguishes utilitarianism from egoism has to do with the scope of the relevant consequences. On the utilitarian view one ought to maximize the overall good — that is, consider the good of others as well as one's own good. The Classical Utilitarians, Jeremy Bentham and John Stuart Mill, identified the good with pleasure, so, like Epicurus, were hedonists about value. They also held that we ought to maximize the good, that is, bring about 'the greatest amount of good for the greatest number'."*

### 3.1.2.2. Texto 2: Ellipse

El texto pertenece a la disciplina de Matemáticas (Geometría Analítica) y procede del libro “Algebra and Trigonometry 2e de OpenStax” [32]. Introduce el concepto de elipse como sección cónica, describe su definición formal y explica su construcción mediante objetos cotidianos.

*“Conic sections can also be described by a set of points in the coordinate plane. Later in this chapter, we will see that the graph of any quadratic equation in two variables is a conic section. The signs of the equations and the coefficients of the variable terms determine the shape. This section focuses on the four variations of the standard form of the equation for the ellipse. An ellipse is the set of all points  $(x,y)$  in a plane such that the sum of their distances from two fixed points is a constant. Each fixed point is called a focus (plural: foci). We can draw an ellipse using a piece of cardboard, two thumbtacks, a pencil, and string. Place the thumbtacks in the cardboard to form the foci of the ellipse. Cut a piece of string longer than the distance between the two thumbtacks (the length of the string represents the constant in the definition). Tack each end of the string to the cardboard, and trace a curve with a pencil held taut against the string. The result is an ellipse.”*

### 3.1.2.3. Texto 3: Homeostasis

El texto pertenece a la disciplina de Biología (Fisiología) y procede de la Wikipedia [33]. Explica que los procesos metabólicos de los organismos sólo pueden desarrollarse en ambientes físico-químicos muy específicos y detalla los mecanismos homeostáticos que regulan la composición del líquido extracelular en mamíferos.

*“The metabolic processes of all organisms can only take place in very specific physical and chemical environments. The conditions vary with each organism, and also with whether the chemical processes take place inside the cell or in the interstitial fluid bathing the cells. The best-known homeostatic mechanisms in humans and other mammals are regulators that keep the composition of the extracellular fluid (or the ‘internal environment’) constant, especially with regard to the temperature, pH, osmolality, and the concentrations of sodium, potassium, glucose, carbon dioxide, and oxygen. However, a great many other homeostatic mechanisms, encompassing many aspects of human physiology, control other entities in the body. Where the levels of variables are higher or lower than those needed, they are often prefixed with hyper- and hypo-, respectively such as hyperthermia and hypothermia or hypertension and hypotension.”*

### 3.1.2.4. Texto 4: Inflation

El texto pertenece a la disciplina de Economía (Macroeconomía) y procede del sitio Tutor2u [34]. Define la inflación por demanda como aquella que se produce cuando la demanda agregada de bienes y servicios supera la capacidad productiva de la economía, y expone sus causas principales: gasto de consumidores, políticas monetarias expansivas, inversión empresarial y gasto público.

*“Demand-pull inflation is a type of inflation that occurs when the overall demand for goods and services in an economy outpaces the economy’s ability to supply them. It happens when the aggregate demand increases faster than the aggregate supply. In demand-pull inflation, there is an increase in consumer spending, business investment, or government expenditure, leading to a surge in demand. As demand rises, businesses may struggle to meet the increased demand with their existing production capacities, resulting in upward pressure on prices. Several factors can contribute to demand-pull inflation: (1) Strong Consumer Spending: When consumers have higher disposable income or confidence in the economy, they tend to spend more on goods and services. Increased consumer demand can lead to a shortage of supply relative to the demand, pushing prices higher. (2) Expansionary Monetary Policy: Central banks may implement expansionary monetary policies, such as lowering interest rates or increasing the money supply, to stimulate economic growth. This can boost borrowing and investment, resulting in increased aggregate demand and potential inflationary pressures. (3) Government Spending: Increased government spending, particularly when financed through deficit spending or borrowing, can stimulate demand in the economy. Government expenditure on infrastructure projects, welfare programs, or defense can contribute to demand-pull inflation. (4) Investment and Business Confidence: When businesses are optimistic about future prospects, they may increase investment and expand their production capacities. However, if the demand for their products outstrips the expanded supply, it can lead to inflationary pressures.”*

## 3.2. Métricas de Evaluación

El objetivo de las métricas de evaluación de grafos es medir cuantitativamente cómo cada enfoque de generación de KG influye en la precisión, la cobertura y la similitud estructural, tanto en el plano léxico como en el semántico. Para ello, se toma el grafo Gold como referencia de calidad. Con el fin de calcular estas métricas, se han creado funciones en Python que, dados dos ficheros CSV correspondientes a dos grafos, calculan automáticamente:

- **Precisión:** : proporción de tripletas generadas que coinciden con las del grafo Gold.
- **Cobertura:** proporción de tripletas del grafo Gold que aparecen en el grafo generado.
- **Similitud estructural:** grado de solapamiento en la topología de ambos grafos (nodos y arcos), medido mediante una adaptación del índice de Jaccard para sus conjuntos de nodos y arcos.

También se calculará F1 score para medir el equilibrio entre precisión y cobertura en una única métrica. Cada una de estas métricas se evalúa en dos niveles: plano semántico y plano léxico.

### Plano semántico

El plano semántico compara el contenido conceptual de las tripletas, considerando sinónimos y equivalencias, de modo que dos tripletas se consideren coincidentes si transmiten el mismo significado aunque difieran léxicamente. Para decidir si dos nodos o dos arcos de ambos grafos deben tratarse como “los mismos”, es necesario detectar sinónimos, equivalencias y variantes estilísticas. Así se pueden unificar entidades o relaciones que, aunque escritas de forma diferente, aluden al mismo concepto (por ejemplo, “human beings” y “humans”).

Para ello se utilizan modelos de embeddings como *Flair* o *Sentence Transformers*. Estos modelos convierten palabras o frases en vectores numéricos en un espacio semántico: cuanto más cercanos estén dos vectores, más similares son sus significados. De este modo, expresiones equivalentes, aunque distintas en su redacción, quedan cerca en ese espacio vectorial y pueden compararse automáticamente.

- En el caso de *Flair* se opta por probar el modelo *FastText Word Embeddings* de 300 dimensiones entrenado sobre el corpus *Common Crawl*, porque ofrece representaciones ligeras, rápidas y fáciles de integrar en pipelines de NLP con recursos limitados [35].
- Para *Sentence Transformers* se prueba el modelo *all-MiniLM-L6-v2*, basado en arquitectura transformer y optimizado para entornos de cómputo restringido, que entrega directamente embeddings de oraciones afinados para tareas de similitud semántica [36].

Tabla 3.1: Comparación de Flair y Sentence Transformers

Par de entidades	Equivalencia	Similitud Flair	Similitud ST	Mejor método
Classical utilitarians - classical Utilitarians	True	0.7892	1.0000	ST
developed in - DEVELOPED_IN	True	0.4602	0.8440	ST
history of philosophy - philosophy history	True	0.3397	0.9739	ST
20th century - century	True	0.5748	0.7549	ST
human beings - humans	True	0.7002	0.8141	ST
position - stance	True	0.5115	0.6278	ST
ethics - morality	True	0.6971	0.6855	Flair
utilitarianism - deontology	False	0.6543	0.1628	ST

En la Tabla 3.1 se observa que en general el modelo Sentence Transformers captura mejor la similitud semántica que Flair, generando puntuaciones más altas en parejas equivalentes y más bajas en términos diferentes. Por ese motivo, se empleará el modelo de Sentence Transformers para identificar equivalencias entre nodos y arcos y aplicar las métricas semánticas de manera más flexible y robusta.

## Plano léxico

El plano léxico compara la igualdad literal de las cadenas que representan sujeto, predicado y objeto en cada tripleta.

Para esta comparación se emplea la distancia de edición de Levenshtein, implementada con la función *distance.edit\_distance* de NLTK, la biblioteca de Python para PLN. Esta función calcula el número mínimo de operaciones (inserciones, eliminaciones o sustituciones) necesarias para convertir una cadena de texto en otra [37][38]. Se fija un umbral máximo de distancia igual a 2, valor seleccionado tras pruebas preliminares, porque tolera variaciones menores de ortografía y uso de mayúsculas (por ejemplo, “Utilitarianism” vs. “utilitarianism”) sin coincidir términos léxicamente diferentes.

A continuación se describen con detalle las métricas de precisión, cobertura y similitud estructural utilizadas para evaluar los grafos.

### 3.2.1. Precisión

Por un lado la precisión semántica mide el porcentaje de nodos y arcos generados que poseen un equivalente conceptual en el grafo Gold. Por otro lado, la precisión léxica cuantifica el porcentaje de esos mismos elementos que coinciden textualmente con los del grafo Gold, permitiendo ligeras variaciones ortográficas o sintácticas según el umbral de Levenshtein.

Para calcular esta métrica, se definen las siguientes fórmulas:

$$\text{NodeAccuracy} = \frac{TP_{nodes}}{TP_{nodes} + FP_{nodes}}$$

$$\text{EdgeAccuracy} = \frac{TP_{edges}}{TP_{edges} + FP_{edges}}$$

$$\text{TotalAccuracy} = \frac{\text{NodeAccuracy} + \text{EdgeAccuracy}}{2}$$

Donde:

- **TP (True Positives)**: número de nodos o arcos generados por el sistema que tienen un equivalente correcto en el grafo Gold, ya sea en términos semánticos o léxicos (según la métrica).
- **FP (False Positives)**: número de nodos o arcos generados por el sistema que no tienen ningún equivalente válido en el grafo Gold, es decir, representan información incorrecta o irrelevante.

### 3.2.2. Cobertura

La cobertura semántica representa la proporción de los elementos del grafo Gold que han sido correctamente capturados mediante correspondencias basadas en el significado. La cobertura léxica indica la proporción de elementos del grafo de referencia que quedan incluidos en el grafo generado al aplicar la comparación aproximada basada en la distancia de Levenshtein.

La cobertura se calcula mediante las siguientes fórmulas:

$$\text{NodeRecall} = \frac{TP_{nodes}}{TP_{nodes} + FN_{nodes}}$$

$$\text{EdgeRecall} = \frac{TP_{edges}}{TP_{edges} + FN_{edges}}$$

$$\text{TotalRecall} = \frac{\text{NodeRecall} + \text{EdgeRecall}}{2}$$

Donde:

- **TP (True Positives)**: número de nodos o arcos del grafo Gold que han sido correctamente identificados en el grafo generado.
- **FN (False Negatives)**: número de nodos o arcos del grafo Gold que no han sido recuperados por el sistema, es decir, que están ausentes en el grafo generado.

### 3.2.3. Similitud estructural

Para cuantificar la semejanza global de la topología de los grafos se emplea una adaptación del índice de Jaccard, utilizado habitualmente para medir la similitud entre conjuntos [39]:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Aplicando un enfoque similar a la estructura del grafo, se propone la siguiente fórmula donde las intersecciones y uniones se realizan tras mapear equivalencias semánticas o léxicas entre entidades y relaciones:

$$\text{Similitud} = \frac{|nodos_{pred} \cap nodos_{gold}| + |arcos_{pred} \cap arcos_{gold}|}{|nodos_{pred} \cup nodos_{gold}| + |arcos_{pred} \cup arcos_{gold}|}$$

El resultado de esta similitud es un valor normalizado entre 0 y 1, donde 0 indica ausencia total de solapamiento estructural y 1 coincidencia perfecta. Al sumar por igual las contribuciones de nodos y arcos, esta métrica penaliza de manera equilibrada las diferencias en cualquiera de los dos componentes, independientemente del tamaño de los grafos. Esta fórmula permitirá evaluar de forma conjunta y equilibrada la coincidencia de nodos y arcos entre los grafos comparados.

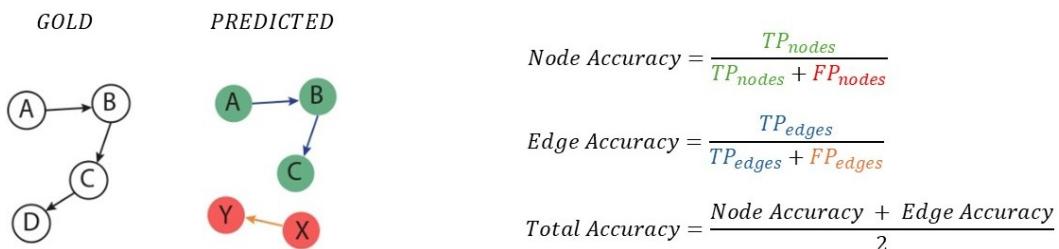


Figura 3.1: Métrica de precisión para grafos

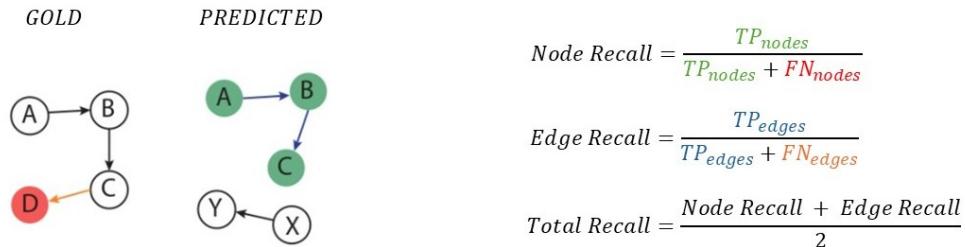


Figura 3.2: Métrica de cobertura para grafos

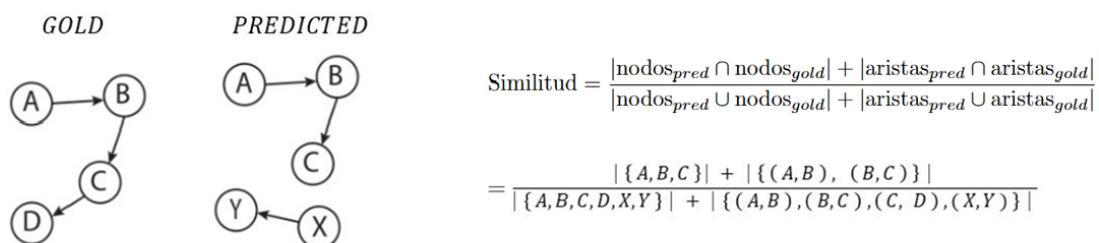


Figura 3.3: Métrica de similitud estructural para grafos

### 3.3. Creación de Grafos de Conocimiento

En esta sección se describen los métodos empleados para generar los cuatro grafos de conocimiento comparados en el estudio.

#### 3.3.1. Grafo Gold

El grafo Gold se genera manualmente para cada texto siguiendo un procedimiento general para generar grafos de conocimiento. Aunque la interpretación puede variar entre evaluadores, a continuación se describen una serie de pasos sistemáticos y replicables que ilustran cómo se podría construir manualmente un grafo de conocimiento:

##### 1. Lectura y comprensión del texto

Realizar una lectura atenta para captar el contenido global, identificar conceptos clave y entender el contexto.

##### 2. Identificación de entidades principales

Extraer del texto los elementos relevantes (personas, conceptos, etc.) que pasarán a ser los nodos del grafo.

##### 3. Determinación de relaciones entre entidades

Analizar cómo se conectan esas entidades (por ejemplo, acciones, asociaciones o pertenencias) y definir los arcos correspondientes.

##### 4. Establecimiento de la dirección de las relaciones

Indicar hacia dónde apunta cada conexión para conservar su significado.

Los grafos generados mediante este proceso manual sirven como referencia de calidad para comparar los resultados automáticos obtenidos con las diferentes herramientas de KGC.

#### 3.3.2. Grafo Baseline con OpenIE

La extracción de tripletas sujeto-predicado-objeto mediante OpenIE constituye el elemento fundamental para generar un grafo de conocimiento con recursos limitados. Para ello, se utiliza el componente OpenIE de Stanford CoreNLP en su configuración predeterminada, al que únicamente se añade un módulo básico de resolución de correferencias para unificar las menciones equivalentes dentro de cada texto.

La resolución de correferencias permite identificar cuándo diferentes expresiones dentro del texto (por ejemplo, pronombres o sustantivos) se refieren a la misma entidad. Esto se representa internamente mediante estructuras llamadas *CorefChain*. Un *CorefChain* es una estructura que agrupa todas las menciones que se refieren a una misma entidad a lo largo de un texto. Antes de guardar cada triplete en el archivo CSV donde se almacenan las tripletas del grafo, se revisa si el sujeto o el objeto corresponde a una mención con correferencia. Si es así, se sustituye el pronombre por el texto original de su referente. CoreNLP proporciona el método *getRepresentativeMention()*, que devuelve la mención más representativa dentro de una cadena de correferencia, priorizando menciones completas o con más contexto [40]. Por ejemplo, en un texto como:

*“Marie Curie was a brilliant scientist. She discovered radium.”*

El sistema identifica que “*She*” se refiere a “*Marie Curie*”, y ambos elementos pertenecen a una misma *CorefChain*. El método *getRepresentativeMention()* en este caso detecta “*Marie Curie*” como la mención más representativa.

Otra opción podría haber sido usar el parámetro *openie.resolve\_coref* de OpenIE. Según la documentación [41] esta opción debería permitir que las menciones pronominales sean sustituidas automáticamente por su forma canónica en las tripletas. Sin embargo, en la práctica no produce ningún efecto sobre las tripletas extraídas, como se ha comprobado empíricamente. Por tanto, para lograr una resolución de correferencias efectiva, es necesario realizar la sustitución manualmente utilizando las estructuras *CorefChain* conforme se van generando las tripletas.

A continuación, se muestra un ejemplo de cómo varían las tripletas cuando se resuelven manualmente las correferencias frente a cuando no se hace:

Sujeto	Predicado	Objeto
Curie	be	brilliant
Marie Curie	be	brilliant scientist
Marie Curie	be	scientist
Marie Curie	discover	radium

Tabla 3.2: Tripletas con resovedor de correferencias

Sujeto	Predicado	Objeto
Curie	be	brilliant
Marie Curie	be	brilliant scientist
Marie Curie	be	scientist
she	discover	radium

Tabla 3.3: Tripletas sin resovedor de correferencias

Como se observa, sin la resolución de correferencias se pierden referencias explícitas a la entidad real, lo que puede dificultar el análisis semántico posterior.

### 3.3.3. Grafo Neo4j LLM Graph Builder

Para generar el grafo con el Neo4j LLM Graph Builder no se ha definido ningún esquema previo, ya que el objetivo es emplear la herramienta de forma general, sin depender de una estructura preestablecida, al igual que ocurre con el sistema basado en OpenIE. Únicamente se ha utilizado un prompt genérico para guiar la extracción:

*“Extract named entities and domain-relevant concepts from the text. Prioritize proper nouns, including people, organizations, locations, historical figures, technical terms, scientific processes, and conceptual entities.”*

Además, no se han aplicado funciones adicionales de mejora del grafo (graph enhancement), con el fin de garantizar una comparación justa entre el grafo generado por Neo4j LLM Knowledge Graph

Builder y el obtenido mediante OpenIE. De este modo, se evalúa la capacidad del Graph Builder para extraer información semántica directamente a partir del texto original, usando únicamente un prompt, sin configuraciones ni tratamientos posteriores.

Para ello, se cargarán únicamente los textos del dataset de referencia y se examinará el grafo de entidades, el cual representa las relaciones semánticas entre conceptos detectados por el sistema. Aunque el modelo de Neo4j también genera conexiones internas entre los distintos chunks de texto y entre estos y sus documentos, se opta por analizar exclusivamente las entidades y relaciones extraídas, descartando dichas conexiones estructurales internas. De este modo, se enfoca el análisis en la calidad de la información semántica capturada.

Así, se generará un grafo utilizando una herramienta respaldada por grandes modelos de lenguaje y recursos computacionales avanzados. Este grafo, junto con el grafo Gold construido manualmente y el generado mediante Notebook LM, actuará como punto de comparación para valorar el rendimiento de la herramienta propuesta para funcionar con recursos limitados.

### 3.3.4. Grafo Notebook LM

Se empleará NotebookLM de manera análoga a como se ha utilizado el Neo4j LLM Knowledge Graph Builder. Se cargarán los archivos de texto del conjunto de datos de referencia y se aplicará el mismo prompt a todos ellos. Hay que aclarar que, a diferencia de Neo4j, NotebookLM no está diseñado específicamente para generar grafos, sino que es una herramienta multifuncional de apoyo a la investigación. No obstante, mediante un prompt cuidadosamente diseñado es posible extraer triplets semánticas útiles para la construcción de un grafo. El prompt utilizado es el siguiente:

*“From the following text, create a knowledge graph by identifying the key entities and the relationships between them. Return the information as a list of subject–predicate–object triples.”*

La salida de NotebookLM consistirá en una lista de triplets sujeto–predicado–objeto, que se almacenará en formato CSV. Este grafo se comparará, mediante las métricas, con los otros tres enfoques considerados en el trabajo: el grafo generado manualmente (grafo Gold), el obtenido con Neo4j LLM Knowledge Graph Builder y el producido por el sistema de extracción de OpenIE.

## 3.4. Bloque I: Expansión de Cobertura

En este bloque se abordará la implementación de dos estrategias destinadas a mejorar la cobertura del grafo Baseline generado por OpenIE:

- **Estrategia 1:** Normalización mediante chunking y re-extracción.
- **Estrategia 2:** Extracción de triplets a partir de chunks verbales.

### 3.4.1. Estrategia 1: Normalización por Chunking y Re-Extracción

#### Objetivo

El objetivo es capturar nuevas relaciones que OpenIE omite o segmenta de forma incompleta, especialmente en las entidades (sujetos y objetos), re-extrayendo triplets una vez que los sintagmas nominales se han convertido en unidades léxicas indivisibles.

#### Hipótesis

Al convertir cada sintagma nominal o chunk en un token único y volver a aplicar el extractor, se incrementará el número de triplets cuyos sujetos y objetos coincidan con entidades completas. Esto aumentará el numero de triplets valiosas del grafo, mejorando su cobertura sin introducir un ruido que degrade de forma significativa su precisión.

#### Motivación

En la extracción inicial con OpenIE muchos sujetos y objetos resultan parcial o incorrectamente segmentados. Por ejemplo, de la frase:

*“Though not fully articulated until the 19th century, proto-utilitarian positions can be discerned throughout the history of ethical theory.”*

OpenIE extrae la tripleta:

(“position”, “fully articulate until”, “century”),

Cuando sería más preciso obtener:

(“proto-utilitarian positions”, “fully articulate until”, “19th century”)

Por tanto, parece viable utilizar alguna técnica que permita crear una serie de chunks completos que sirvan como base para guiar una segunda extracción de OpenIE que aporte nueva información.

#### Descripción de la estrategia

En primer lugar, se realiza la extracción de noun chunks. Para ello se utiliza Stanford CoreNLP, que analiza la estructura sintáctica de cada oración y etiqueta automáticamente los sintagmas nominales (NP). El resultado se almacena en un fichero JSON que asocia cada chunk a su oración de origen. Por ejemplo, para la frase:

*“Utilitarianism is one of the most powerful and persuasive approaches to normative ethics in the history of philosophy.”*

Se extraen los siguientes conceptos:

*“Utilitarianism”, “normative ethics”, “normative ethics in the history of philosophy”, “the history of philosophy”*

Para la frase:

*“The metabolic processes of all organisms can only take place in very specific physical and chemical environments.”*

Los chunks identificados incluyen:

*“The metabolic processes”, “The metabolic processes of all organisms”, “very specific physical and chemical environments”*

Se detectan varios problemas en los chunks generados por CoreNLP. Con el fin de mejorar su calidad y corregir las deficiencias identificadas, se aplican tres transformaciones sucesivas:

1. **Filtrado de chunks excesivamente largos o con verbos:** Se identifican y excluyen de la colección aquellos sintagmas que contienen verbos, pues no representan conceptos nominales puros que puedan servir como entidades del grafo.
2. **Enriquecimiento con chunks de SpaCy:** Sobre el mismo texto, se extraen noun chunks con SpaCy, que tiende a ofrecer entidades más específicas y depuradas. A continuación, se fusiona este nuevo conjunto con los chunks resultantes del paso anterior. Por ejemplo, para la oración:

*“One thing to note is that the theory is a form of consequentialism: the right action is understood entirely in terms of consequences produced.”*

SpaCy extrae los siguientes sintagmas:

*“One thing”, “the theory”, “a form”, “consequentialism”, “the right action”, “terms”, “consequences”*

En cambio, CoreNLP generaba versiones más largas y menos precisas, como:

*“a form of consequentialism”, “consequences produced” o “terms of consequences produced”.*

La fusión garantiza un repertorio ampliado, en el que SpaCy aporta la granularidad que a CoreNLP le falta.

3. **Limpieza de stop words iniciales:** Se eliminan únicamente las stop words que aparecen al comienzo de cada chunk, preservando las stop words interiores, ya que pueden formar parte de términos compuestos relevantes (p. ej. “terms of consequences”). Por ejemplo, a partir de:

*“There”, “this general claim”*

Se obtiene:

*“general claim”*

Una vez completadas estas transformaciones, se dispone de un conjunto de chunks más representativo y manejable.

Dado que el módulo de OpenIE de Standford CoreNLP no permite especificar sobre qué conceptos extraer triplets, se aprovecha el conjunto de noun chunks limpiados y enriquecidos para guiar una segunda pasada. En primer lugar, se aplica la normalización por chunking, en la que cada sintagma nominal del texto de referencia se convierte en una unidad léxica indivisible uniendo sus palabras con guiones bajos. Por ejemplo:

“proto utilitarian positions” → “proto\_utilitarian\_positions”  
“19th century” → “19th\_century”  
“history of ethical theory” → “history\_of\_ethical\_theory”

A continuación, sobre el texto así transformado se realiza la re-extracción de triplets con OpenIE. De este modo, el extractor trata las entidades compuestas como tokens únicos y puede capturar relaciones que antes quedaban fragmentadas o pasaban inadvertidas. Si se generan nuevas triplets, estas se incorporan al grafo con el objetivo de enriquecer su cobertura de entidades completas.

### 3.4.2. Estrategia 2: Extracción de Triplets de Chunks con Verbos

#### Objetivo

Capturar relaciones adicionales extrayendo triplets directamente de los chunks que incluyen verbos, de modo que el grafo incorpore nuevas conexiones y aumente su cobertura.

#### Hipótesis

Al reutilizar los chunks verbales, separados previamente para evitar extracciones de entidades erróneas, como entradas del extractor de triplets, se capturarán relaciones no detectadas en la extracción inicial. Esto incrementará la cobertura del grafo sin añadir ruido significativo.

**Motivación** Durante la limpieza de chunks (Estrategia 1) se aislan aquellos que contienen verbos ya que no representan conceptos nominales puros. Sin embargo, estos sintagmas describen acciones o relaciones implícitas en el texto y constituyen una fuente potencial de triplets valiosas que OpenIE no captura al generar el Baseline.

#### Descripción de la estrategia

Se recogen todos los chunks marcados como verbales y se procesan con el extractor de triplets de OpenIE como oraciones independientes. Por ejemplo, del chunk “interstitial fluid bathing the cells” se espera extraer una tripla similar a (“interstitial fluid”, “bathing”, “cells”), que en la extracción inicial sobre el texto no se identificó. Estas nuevas relaciones se incorporan al grafo con el objetivo de aprovechar la información verbal de estos chunks y aumentar el número total de triplets valiosas extraídas.

## 3.5. Bloque II: Homogeneización y Refinamiento

En este bloque se abordará la implementación de tres estrategias destinadas a refinar predicados mal extraídos y homogeneizar entidades del grafo.

- Estrategia 3: Corrección de verbos modales
- Estrategia 4: Mapeo de sujetos y objetos a chunks de CoreNLP
- Estrategia 5: Mapeo de sujetos y objetos con chunks limpiados y enriquecidos (CoreNLP + SpaCy)

### 3.5.1. Estrategia 3: Corrección de Verbos Modales

#### Objetivo

Se pretende generar predicados más informativos y objetos más puros reubicando los verbos modales que se duplican en predicado y objeto.

#### Hipótesis

Al trasladar los modales (“can”, “may”, “should”, “will”, etc.) y sus complementos al propio predicado, y descartar las tripletas que pierdan su objeto, se obtendrá un conjunto de relaciones con predicados más completos y objetos más limpios, incrementando la precisión y la calidad del grafo.

#### Motivación

En los grafos Baseline aparece a menudo el mismo verbo modal repetido como predicado y como inicio de objeto, por ejemplo:

(“position”, “can”, “can discern”)  
 (“position”, “can”, “can discern throughout history”)  
 (“conic section”, “can”, “can describe by set of points in coordinate plane”)

Estos duplicados fragmentan la relación y ensucian el objeto con palabras auxiliares que no aportan significado propio.

#### Descripción de la estrategia

Se define primero la lista de modales a corregir (“can”, “could”, “may”, “might”, etc). Para cada tripleta, si el predicado coincide con uno de estos modales, se analiza el objeto palabra a palabra hasta localizar el primer término de categoría nominal. Todos los tokens anteriores a ese término se concatenan al modal para formar un predicado ampliado, y los tokens restantes constituyen el nuevo objeto. Si tras esta operación el objeto queda vacío, la tripleta se elimina. Por ejemplo:

- La tripleta (“position”, “can”, “can discern”) se descarta,
- Mientras que (“position”, “can”, “can discern throughout history”) se transforma en (“position”, “can discern throughout”, “history”)

Este mecanismo unifica el verbo modal con la acción principal y limpia el objeto, pudiendo obtener relaciones más precisas y reducir la redundancia en el grafo.

### 3.5.2. Estrategia 4: Mapeo de Sujetos y Objetos a Chunks de CoreNLP

#### Objetivo

Se sustituirán entidades fragmentadas en el grafo Baseline por sintagmas nominales extraídos automáticamente. Para ello, se mapearán sujetos y objetos a dichos sintagmas, con el fin de obtener una representación más coherente y alineada con la estructura gramatical del texto original.

#### Hipótesis

Al sustituir entidades fragmentadas por sintagmas nominales gramaticalmente sólidos, se espera obtener una representación más limpia y más alineada con el contenido semántico del texto original, aumentando así la precisión del grafo.

#### Motivación

En lingüística computacional, los noun chunks son secuencias de palabras que desempeñan una función sustantiva dentro de una oración. Suelen corresponderse con conceptos gramatical y semánticamente coherentes, por lo que resultan adecuados para representar entidades en grafos de conocimiento. A diferencia de algunos sujetos y objetos generados por sistemas de extracción como OpenIE, que pueden cortar o alterar la estructura de los términos, los chunks tienden a ser una representación más precisa de las entidades.

#### Descripción de la estrategia

Se utilizan los chunks extraídos por CoreNLP durante la Estrategia 1, sin aplicarles ninguna limpieza por el momento.

A continuación, se procede al mapeo de cada sujeto u objeto de las tripletas generadas por OpenIE únicamente contra los chunks extraídos de la misma oración. Para ello, se lematiza tanto la entidad original como cada chunk, comparando sus lemas. Se considera que coinciden si todos los lemas de la entidad están incluidos en los lemas del chunk. En caso de que una entidad corresponda a varios chunks, se selecciona el que aporte la menor cantidad de lemas adicionales. Finalmente, cuando se identifica un chunk adecuado, la entidad original se reemplaza por ese chunk. Si no existe correspondencia, se conserva la forma original.

Este proceso permite normalizar sujetos y objetos sin forzar emparejamientos incorrectos. Por ejemplo, la tripleta (bank, may implement, policy) pasaría a tener sujeto “Central banks” y objeto “Expansionary Monetary Policy”, mientras que (sum, be, constant) se transformaría en sujeto “the sum” y objeto “a constant”.

### 3.5.3. Estrategia 5: Mapeo de Sujetos y Objetos con Chunks Limpiados y Enriquecidos (CoreNLP + SpaCy)

#### Objetivo

El objetivo es refinar y ampliar los noun chunks extraídos con el fin de mapear sujetos y objetos de las tripletas a sintagmas más precisos, para aumentar el número de emparejamientos y mejorar la coherencia semántica de las entidades.

#### Hipótesis

Mejorar la calidad de los chunks y flexibilizar el proceso de mapeo incrementará el número de asignaciones correctas de sujetos y objetos, lo que se traducirá en una mayor limpieza y precisión del grafo. La mejora de la calidad de los chunks consistirá en limpiar stop words, eliminar chunks que contengan verbos y añadir los sintagmas nominales extraídos por SpaCy.

#### Motivación

En la extracción inicial de conceptos con CoreNLP se han identificado varios problemas que justifican la limpieza y el enriquecimiento de los chunks. Por un lado, existen sintagmas excesivamente largos o que incluyen verbos (“the cardboard to form the foci of the ellipse”, “a type of inflation that occurs when the overall demand for goods...”), los cuales no aportan valor semántico como entidades y pueden inducir mapeos erróneos. Por otro, CoreNLP genera chunks genéricos o adornados (“a form of consequentialism”, “terms of consequences produced”) cuando en ocasiones puede resultar preferible extraer conceptos más específicos (“consequentialism”, “consequences”). Finalmente, muchos chunks comienzan con stop words (“the sum”, “a constant”), lo cual no tiene sentido para un grafo de conocimiento.

#### Descripción de la estrategia

A continuación, se aplica un mapeo de sujetos y objetos más flexible. A diferencia del mapeo de la Estrategia 4, este proceso comienza intentando una coincidencia exacta de lemas entre la entidad y los chunks. Si no se encuentra ninguna correspondencia exacta, se selecciona el chunk que comparta con la entidad el mayor número de lemas, incluso cuando no los contenga todos. Este enfoque aporta flexibilidad sin perder coherencia gramatical, asegurando que solo se sustituyan entidades por chunks verdaderamente representativos.

## 3.6. Bloque III: Mejora de Precisión

En este bloque se abordará la implementación de dos estrategias destinadas a mejorar la precisión del grafo.

- Estrategia 6: Filtrado de tripletas por confianza
- Estrategia 7: Eliminación de tripletas similares

### 3.6.1. Estrategia 6: Filtrado de Tripletas por Confianza

#### Objetivo

Se eliminarán todas las tripletas extraídas por OpenIE cuya confianza (`triple.confidence`) sea inferior a 1.0. El objetivo es reducir el ruido estructural del grafo y asegurar que únicamente se mantengan relaciones semánticamente sólidas.

#### Hipótesis

Se plantea que las tripletas con puntuaciones de confianza bajas suelen corresponder a relaciones ambiguas, incorrectas o mal formadas. Se espera que su eliminación incremente la precisión global del grafo sin comprometer sustancialmente la cobertura de relaciones extraídas.

#### Motivación

Al estudiar los cuatro textos de referencia (Utilitarianism, Ellipse, Homeostasis e Inflation), se observa que la mayoría de las tripletas extraídas por OpenIE tienen confianza igual a 1.0. Esto sugiere que un filtrado, centrado en conservar únicamente las tripletas con confianza máxima, podría aplicarse sin comprometer en exceso el número total de relaciones extraídas.

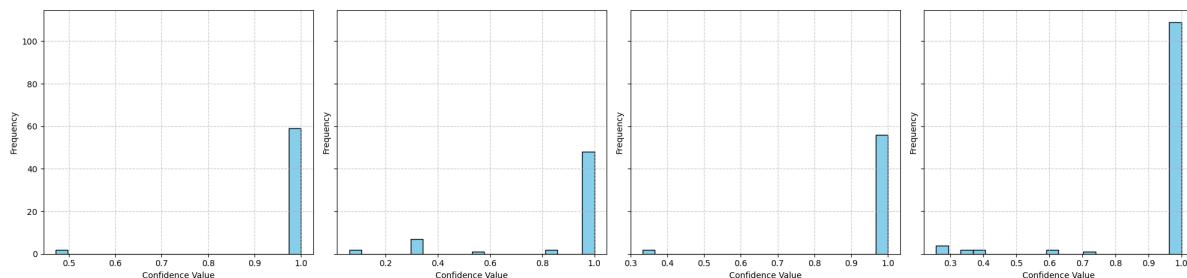


Figura 3.4: Distribuciones de confianza de las tripletas de los grafos Baseline

Se analizan las tripletas de baja confianza:

- En Utilitarianism aparecen relaciones razonables como (`John Stuart Mill, be, hedonist`) con confianza 0.472, pero también tripletas semánticamente débiles.
- En Ellipse surgen tripletas claramente defectuosas, p. ej. (`fix point, be, call`) con confianza 0.061 o (`ellipse two thumbtack, use, piece of cardboard`) con 0.333.
- En Homeostasis y Inflation hay casos de segmentación imprecisa como (`consumer, high disposable income in, economy`) con confianza 0.257.

Estos ejemplos confirman que algunas de las tripletas de baja confianza son ambiguas o están mal formadas.

#### **Descripción de la estrategia**

Para llevar a cabo este filtrado, se extrae en cada tripleta el valor *triple.confidence*, derivado de un *affinity score* que mide la plausibilidad gramatical de la relación [42]. A continuación, se descartan todas aquellas tripletas cuyo valor de confianza sea menor que 1.0.

### **3.6.2. Estrategia 7: Eliminación de Tripletas Similares**

#### **Objetivo**

Se reducirá la redundancia en el grafo eliminando aquellas tripletas que sean idénticas o muy parecidas entre sí, de modo que solo permanezcan las relaciones únicas y más completas.

#### **Hipótesis**

Se espera que la depuración de tripletas redundantes aumente la precisión del grafo, al evitar la sobre-representación de las mismas relaciones y concentrar la información en un conjunto más limpio.

#### **Motivación**

OpenIE tiende a generar múltiples tripletas con significados equivalentes o muy próximos. Por ejemplo:

(“increase”; “be in”; “demand pull inflation”)  
 (“increase”, “be in”, “pull inflation”)  
 (“increase”, “be in”, “inflation”)

Mantener todas esas relaciones en el grafo introduce ruido y puede empeorar las métricas.

#### **Descripción de la estrategia**

Para depurar las tripletas, en primer lugar se eliminan los duplicados exactos, descartando aquellas relaciones en las que sujeto, predicado y objeto coinciden de forma idéntica tras las distintas fases de mapeo o enriquecimiento. A continuación, se agrupan las tripletas que comparten sujeto y objeto. Dentro de cada conjunto, se conserva únicamente la versión cuyo predicado sea más completo cuando uno aparezca como subcadena de otro. Después, se agrupan las tripletas que comparten sujeto y predicado, de modo que, si un objeto está contenido en otro más informativo (por ejemplo, “inflation” frente a “demand pull inflation”), se elimina la tripleta con el objeto más breve, manteniendo solo aquella con el término más representativo. Finalmente, se hace lo mismo agrupando por predicado y objeto.

# Capítulo 4

## Resultados

### 4.1. Evaluación Preliminar: Baseline frente a otros enfoques

En esta sección se presentan los resultados de la comparación entre el grafo Baseline, generado mediante el sistema OpenIE sin aplicar ningún tipo de limpieza ni transformación posterior, y los otros tres enfoques considerados en este trabajo: el grafo Gold, el grafo generado mediante Neo4j LLM Knowledge Graph Builder, y el grafo generado a través de NotebookLM.

El objetivo de esta primera comparación es establecer una línea base cuantitativa sobre la cual evaluar las distintas transformaciones que se aplican posteriormente al grafo generado por OpenIE.

Las métricas descritas en la metodología permiten evaluar no sólo cuánto se parece el grafo generado al grafo Gold en términos estructurales, sino también qué proporción del contenido se ha recuperado (cobertura) y cómo de precisos son los elementos generados (precisión), en los niveles semántico y léxico.

Es importante recordar que, aunque se utiliza un grafo como referencia, el grafo Gold, este no representa una verdad absoluta, sino una posible interpretación válida del contenido.

A continuación, se presentan los resultados obtenidos para el Baseline y los demás enfoques, expresados mediante las métricas descritas anteriormente.

#### Utilitarianism

Los resultados muestran un rendimiento significativamente superior de los enfoques basados en modelos de lenguaje (Neo4jLLM y NotebookLM) frente al Baseline. El grafo generado por NotebookLM presenta el perfil más equilibrado: alcanza la mayor cobertura tanto en el espacio semántico como léxico, así como la mayor similitud estructural con el grafo Gold. Esto indica que es el sistema que más contenido relevante logra capturar, manteniendo una representación estructural coherente.

Por su parte, Neo4jLLM obtiene una precisión más alta en el plano semántico, lo cual sugiere que, aunque extrae menos información, lo hace de manera más precisa y con menor ruido. Sin embargo, su cobertura es más limitada, con valores inferiores a los de NotebookLM.

El grafo generado mediante OpenIE presenta puntuaciones significativamente más bajas en todas las métricas. Aunque contiene un mayor número de nodos y relaciones (37 nodos y 57 arcos frente a los 19 nodos del grafo Gold), parece que gran parte de esa información resulta redundante o poco informativa. Esto podría deberse a la presencia de predicados genéricos como “be” o “can”, así como a la escasa homogeneidad de entidades.

En la Figura 4.1 se muestra mediante un gráfico de araña cómo NotebookLM domina en cobertura y similitud global, mientras que Neo4jLLM destaca en precisión. En conjunto, los resultados sugieren que los sistemas basados en LLM representan una mejora sustancial frente al grafo Baseline, acercándose de forma más efectiva a la estructura del grafo Gold.

#### Ellipse

Los resultados cuantitativos obtenidos para el texto sobre la elipse reflejados en la Figura 4.2 refuerzan los patrones observados previamente. El grafo generado por NotebookLM vuelve a ser el más destacado

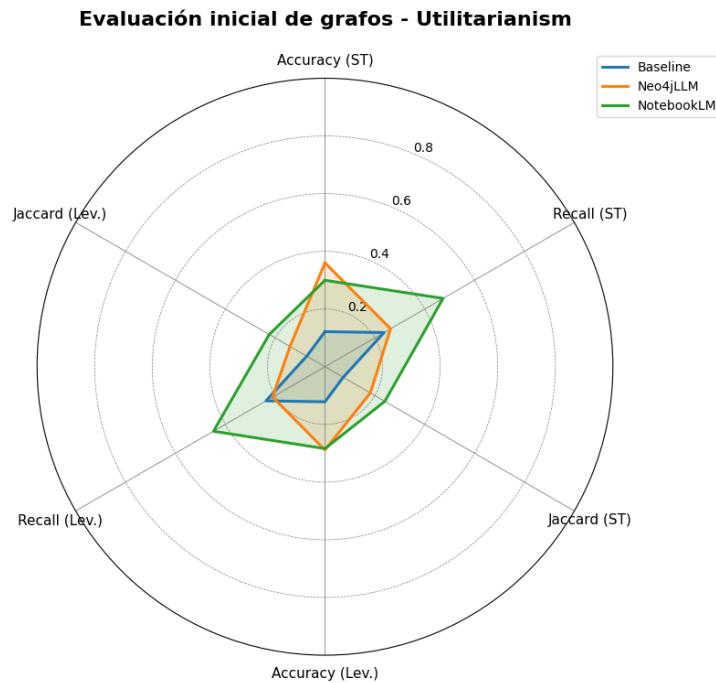


Figura 4.1: Evaluación inicial de grafos - Utilitarianism

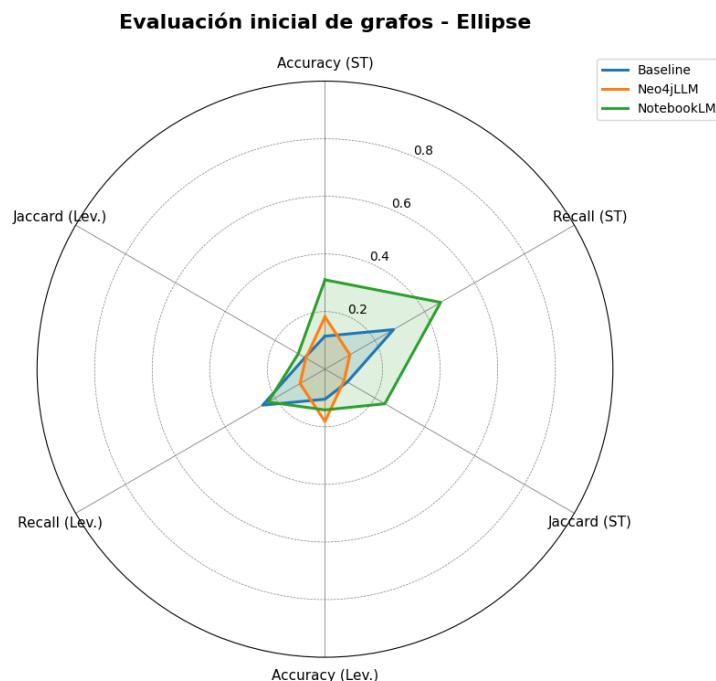


Figura 4.2: Evaluación inicial de grafos - Ellipse

en la mayoría de métricas, alcanzando valores especialmente altos en el plano semántico.

El grafo producido por Neo4jLLM, aunque conceptualmente compacto y más preciso que el Baseline, presenta una cobertura muy limitada. Este rendimiento más bajo se explica por su reducido tamaño: apenas 11 nodos y 7 relaciones, lo que limita significativamente su similitud con el Gold en términos estructurales.

El Baseline generado por OpenIE presenta un rendimiento intermedio: obtiene una cobertura razonable, pero con baja precisión y baja similitud estructural con el Gold. El análisis cualitativo evidencia que este grafo contiene múltiples tripletas redundantes o mal normalizadas, como “can describe by set” y “can describe by set of point”, lo que afecta a la calidad del grafo.

En conjunto, los resultados muestran que NotebookLM genera una representación más equilibrada y estructuralmente rica del contenido sobre la elipse. Aunque Neo4jLLM ofrece precisión en sus predicciones, su baja cobertura limita su efectividad como grafo de conocimiento. En el Baseline, por su parte, se evidencia la necesidad de implementar mejoras sustanciales en limpieza y consolidación de entidades para competir con los enfoques basados en LLMs.

### Homeostasis

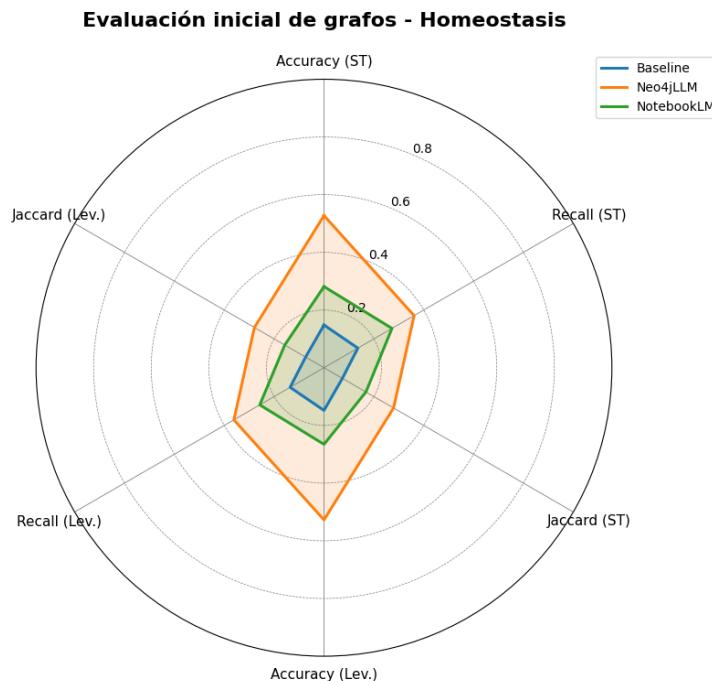


Figura 4.3: Evaluación inicial de grafos - Homeostasis

Los resultados para el texto sobre Homeostasis mostrados en la Figura 4.3 muestran una diferencia clara entre los enfoques evaluados. En este caso, el sistema Neo4jLLM destaca con la mejor puntuación en todas las métricas. Esto refleja un grafo más alineado con el Gold tanto en contenido como en estructura.

El grafo generado por NotebookLM presenta un perfil más equilibrado, pero de menor calidad que el de Neo4j. Aun así, su desempeño es claramente superior al del Baseline en todas las métricas.

El grafo Baseline, como en los casos anteriores, presenta un rendimiento notablemente inferior. Su precisión y recall no superan el 0.15, y su estructura es redundante y poco informativa, con predicados genéricos y relaciones fragmentadas. Esta representación limita la interpretación semántica del contenido y su similitud con el grafo Gold.

Desde el punto de vista cualitativo, el grafo de Neo4jLLM presenta definiciones concisas sobre funciones fisiológicas, con especial énfasis en el medio extracelular. NotebookLM, por su parte, ofrece una representación más extensa que incluye procesos reguladores, pero con menor exactitud formal. El Baseline, como en casos anteriores, presenta muchos elementos superficiales o redundantes.

Para este texto del dataset de referencia, Neo4jLLM ofrece la representación más precisa y estructuralmente coherente, por encima de NotebookLM, quedando el grafo Baseline por debajo en todos los ámbitos.

### Inflation

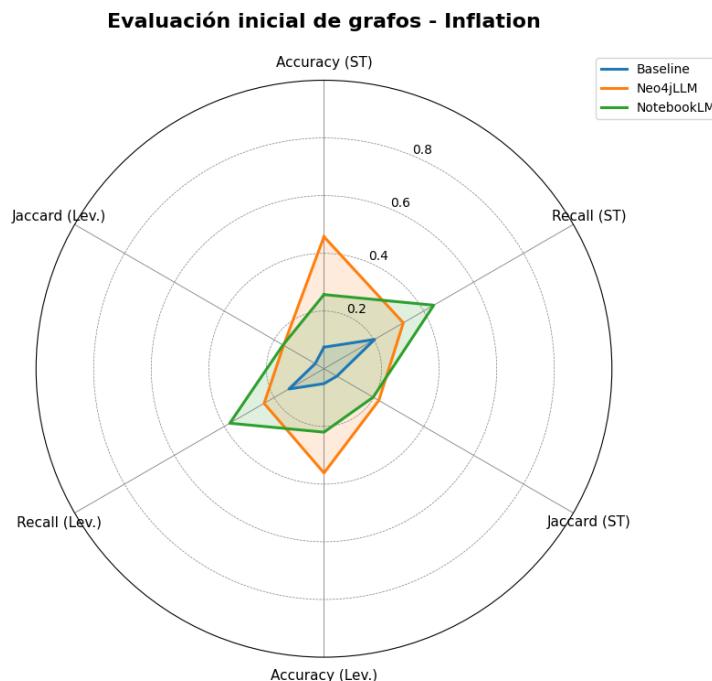


Figura 4.4: Evaluación inicial de grafos - Inflation

Los resultados obtenidos para el texto sobre la inflación refuerzan la tendencia observada en los textos anteriores. Los enfoques basados en modelos de lenguaje (Neo4jLLM y NotebookLM) superan con claridad al grafo Baseline en todas las métricas.

En este caso, NotebookLM alcanza los mayores valores en cobertura, lo que indica que logra capturar una mayor proporción del contenido presente en el grafo Gold. Sin embargo, Neo4jLLM obtiene una mayor precisión, tanto semántica como léxica, siendo el sistema que menos ruido introduce.

En términos de similitud estructural, ambos enfoques de LLMs obtienen resultados muy similares: 0.199 para NotebookLM y 0.206 para Neo4jLLM, lo que refleja que ambos se acercan lo mismo a la representación estructural del grafo Gold.

Por el contrario, el Baseline vuelve a mostrar un rendimiento deficiente, con valores bajos en todas las métricas, especialmente en precisión y similitud estructural. Este comportamiento se debe, como en otros textos, a la abundancia de tripletas redundantes pero escritas de distintas maneras, lo cual genera una estructura repetitiva y conceptualmente ambigua.

En conjunto, nuevamente los enfoques basados en grandes modelos de lenguaje ofrecen representaciones significativamente más precisas y completas que el sistema base. Neo4jLLM destaca por su precisión, mientras que NotebookLM sobresale en cobertura, lo que refleja dos estrategias complementarias para la construcción de grafos de conocimiento a partir de texto.

### Conclusión comparación inicial

En todos los casos del dataset de referencia analizados, los enfoques basados en grandes modelos de lenguaje, y en particular NotebookLM, superan de manera consistente al sistema basado en OpenIE en todas las métricas evaluadas: precisión, cobertura y similitud estructural, tanto en su versión semántica como léxica. Este rendimiento superior se traduce en grafos más alineados con el contenido del grafo Gold y conceptualmente más coherentes. Este resultado es coherente con lo esperado, dado que OpenIE realiza una extracción superficial sin tener en cuenta el contexto semántico, mientras que los LLMs pueden inferir, sintetizar y estructurar relaciones más significativas a partir del contenido textual.

No obstante, es importante tener en cuenta que ninguna métrica cuantitativa por sí sola, incluida la similitud estructural, puede capturar completamente la validez conceptual de un grafo de conocimiento. Dado que el grafo Gold representa solo una posible interpretación del texto, un grafo diferente pero bien construido puede obtener puntuaciones bajas si adopta una estructura igualmente válida pero divergente. Por ello, una evaluación justa y completa requiere complementar el análisis cuantitativo con una revisión cualitativa de los grafos generados.

El análisis cualitativo confirma las tendencias observadas: los grafos generados por LLMs son más coherentes, conceptualmente válidos y estructuralmente más claros que los del sistema OpenIE. Los principales problemas detectados en el Baseline incluyen predicados genéricos, redundancia de tripletas, fragmentación semántica y ambigüedad en las relaciones. En cambio, los grafos generados por LLMs, aunque más compactos, tienden a representar de forma más precisa y comprensible las relaciones clave del contenido.

También se observan limitaciones en los enfoques con LLMs, especialmente en cuanto a granularidad. Sus grafos suelen contener menos nodos y relaciones que el grafo Gold, lo que sugiere que algunos detalles específicos del texto original pueden perderse en favor de una representación más sintetizada.

En conjunto, estos resultados en la comparación inicial establecen un punto de partida claro para optimizar la calidad del grafo Baseline. Confirman que las prioridades iniciales deberían centrarse en tratar de aumentar la cobertura, mejorar la especificidad de los predicados y minimizar la ambigüedad y en reducir la redundancia, con el objetivo de acercarse tanto a la claridad estructural de los grafos generados por *LLMs* como a la riqueza conceptual del grafo Gold.

## 4.2. Resultados por Bloque de Estrategias

### 4.2.1. Bloque I: Expansión de Cobertura

De acuerdo a la primera carencia que se vio en los grafos Baseline, el objetivo de este bloque de estrategias era capturar nuevas tripletas que incrementen la cobertura global, incorporando relaciones que el extractor inicial no detectó.

#### 4.2.1.1. Estrategia 1: Normalización por Chunking y Re-Extracción

La idea de partida consistía en convertir cada sintagma nominal de los textos de entrada en un token único y luego volver a aplicar el extractor de triplets, con el objetivo de aumentar el número de relaciones cuyos sujetos y objetos correspondan a entidades completas. Se esperaba así mejorar la cobertura del grafo sin sacrificar demasiado la precisión.

Al aplicar esta estrategia a cada uno de los textos de referencia, se observa un incremento notable en el número de triplets extraídas:

- En *Utilitarianism*, el conteo pasó de 52 a 63 triplets (11 triplets más, +21,2 %).
- En *Ellipse*, aumentó de 48 a 60 triplets (12 triplets más, +25,0 %).
- En *Homeostasis*, subió de 34 a 58 triplets (24 triplets más, +70,6 %).
- En *Inflation*, creció de 101 a 120 triplets (19 triplets más, +18,8 %).

Muchas de estas nuevas relaciones corresponden a triplets que antes estaban fragmentadas o habían sido omitidas porque el extractor de OpenIE no las había reconocido como entidades compuestas. El siguiente paso es verificar si estas triplets adicionales son realmente correctas y si aportan nueva información. Para ello se analiza las métricas de precisión, cobertura, F1 y similitud estructural frente al grafo Gold.

#### Análisis por texto

Para el texto del Utilitarismo, en la Figura 4.5 se aprecia que todas las métricas mejoran, especialmente las coberturas. Tanto la cobertura léxica como la semántica aumentan en un 20 % aproximadamente. Estos resultados demuestran que al aplicar la Estrategia 1 al grafo Baseline del Utilitarismo, se han extraído triplets de alto valor, que mejoran tanto la precisión, como la cobertura del grafo.

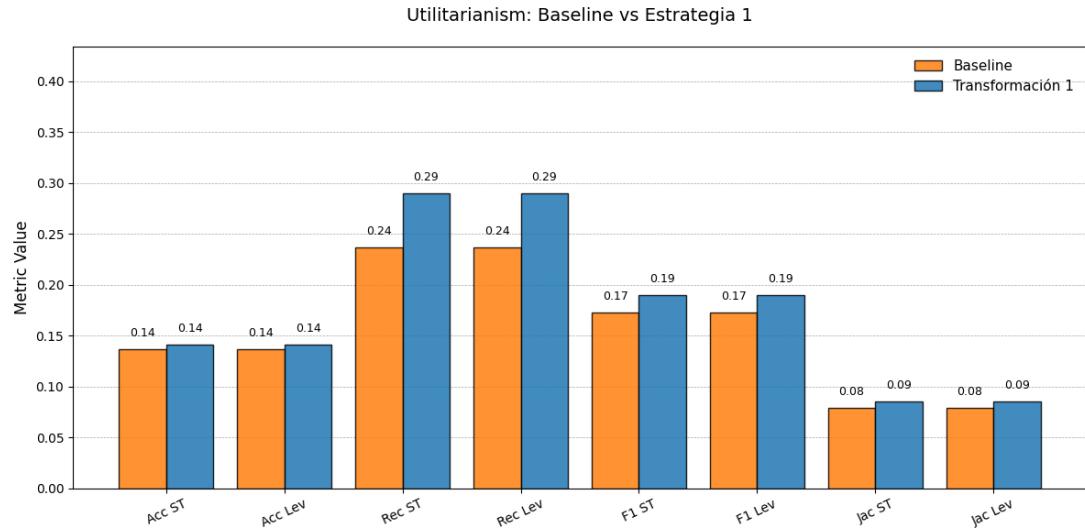


Figura 4.5: Utilitarianism: Baseline vs Estrategia 1

En el caso de *Ellipse*, la Figura 4.6 muestra que las coberturas también aumentan. La cobertura semántica se eleva en un 35.7% y la cobertura léxica en un 12%. Aunque la precisión semántica mejora, la precisión léxica disminuye en un 10%. Los resultados del F1 Score también reflejan esto, mejorando semánticamente, pero empeorando léxicamente. Esto puede deberse a que la Estrategia 1 genera nuevas variantes para referirse a una misma entidad, y esas formas alternativas se alejan más de las expresiones utilizadas en el grafo Gold. Aun así, la mejora en la precisión semántica indica que el significado de las nuevas tripletas sí que es preciso para el texto de la elipse. Esta caída en la precisión léxica se refleja también en una leve reducción de la métrica léxica de similitud estructural.

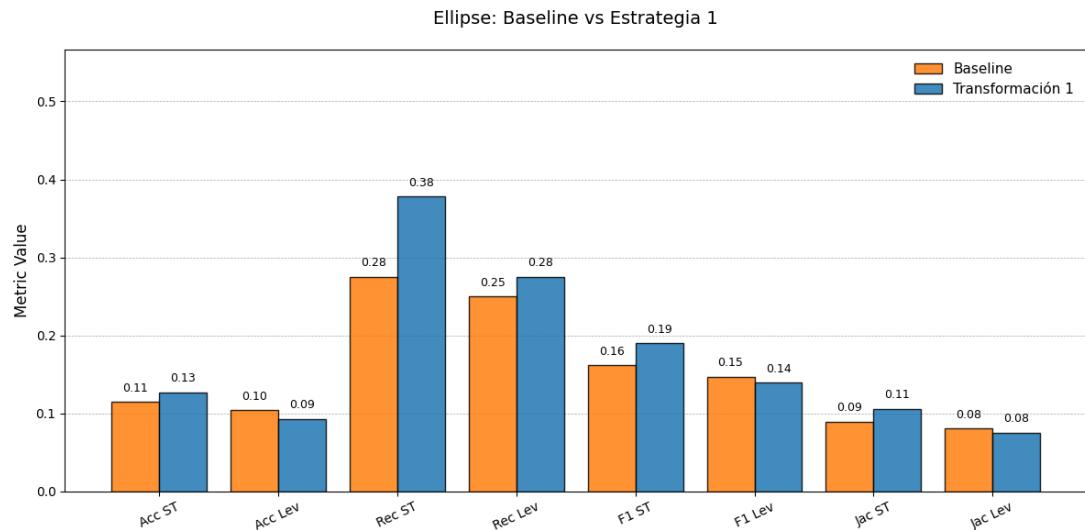


Figura 4.6: Ellipse: Baseline vs Estrategia 1

En el texto de *Homeostasis*, la Estrategia 1 resulta menos efectiva (Figura 4.7). Aunque la cobertura semántica aumenta en un 21%, la cobertura léxica permanece sin cambios y las precisiones caen. Esto

indica que las nuevas entidades extraídas son realmente adicionales, por eso sube la cobertura, pero no coinciden suficientemente con las del grafo Gold, lo que penaliza las precisiones. Esta pérdida de precisión se refleja también en un ligero empeoramiento de la similitud estructural del grafo con el grafo Gold, tal como muestran las métricas de Jaccard.

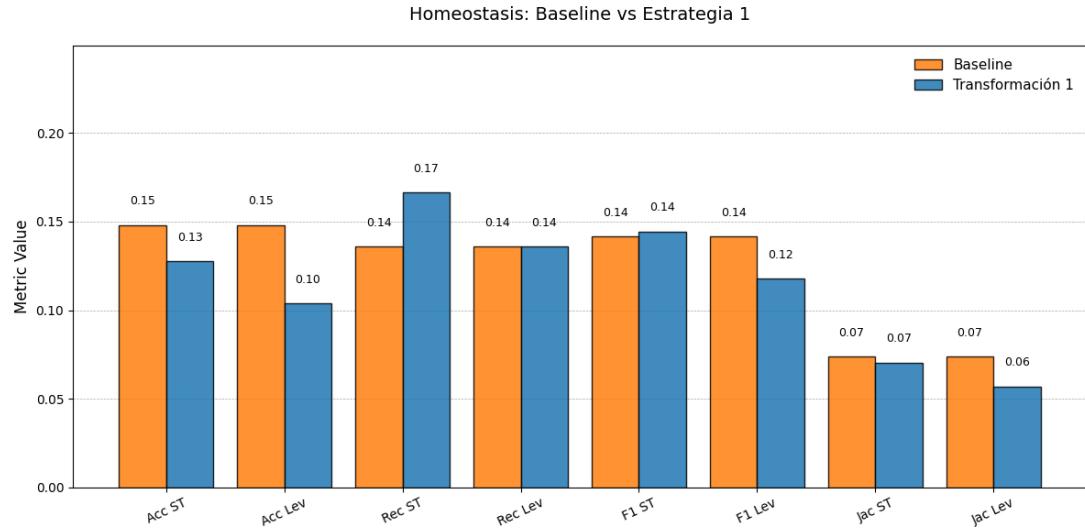


Figura 4.7: Homeostasis: Baseline vs Estrategia 1

En el caso de *Inflation* (Figura 4.8), la cobertura semántica se incrementa en un 24 % y la cobertura léxica en un 13 %, mientras que las precisiones se mantienen prácticamente constantes. La calidad del grafo generado con la Estrategia 1 es similar a la del Baseline, pero incorpora nuevas tripletas valiosas, sobre todo desde el punto de vista semántico.

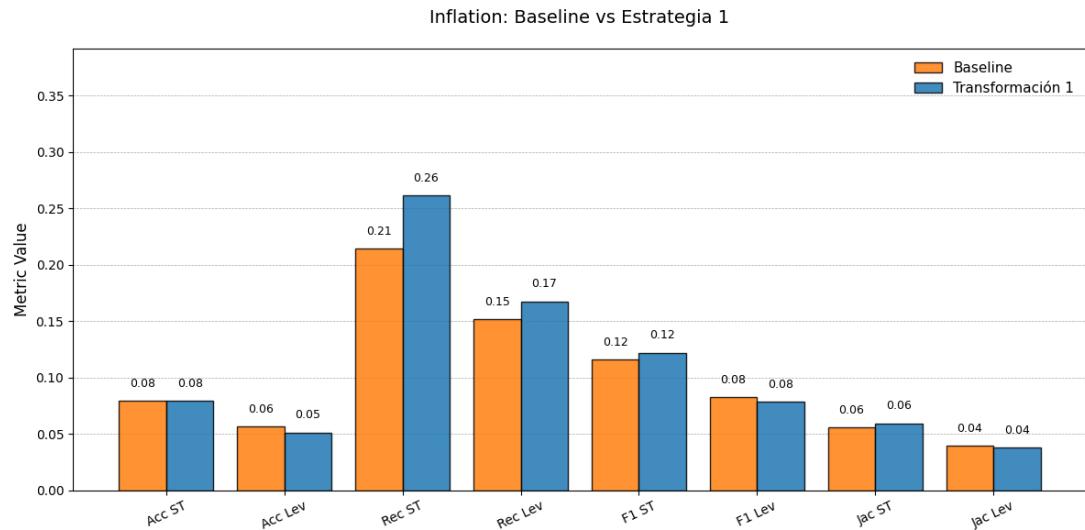


Figura 4.8: Inflation: Baseline vs Estrategia 1

En todos los casos, la re-extracción a partir de chunks unificados mejora la cobertura del grafo sin sacrificar de forma significativa ni la precisión ni la similitud global con el grafo Gold.

### Evolución media de las métricas

Tras aplicar la Transformación 1 al grafo Baseline de los cuatro textos, la evolución media de las métricas queda reflejada en la Figura 4.9.

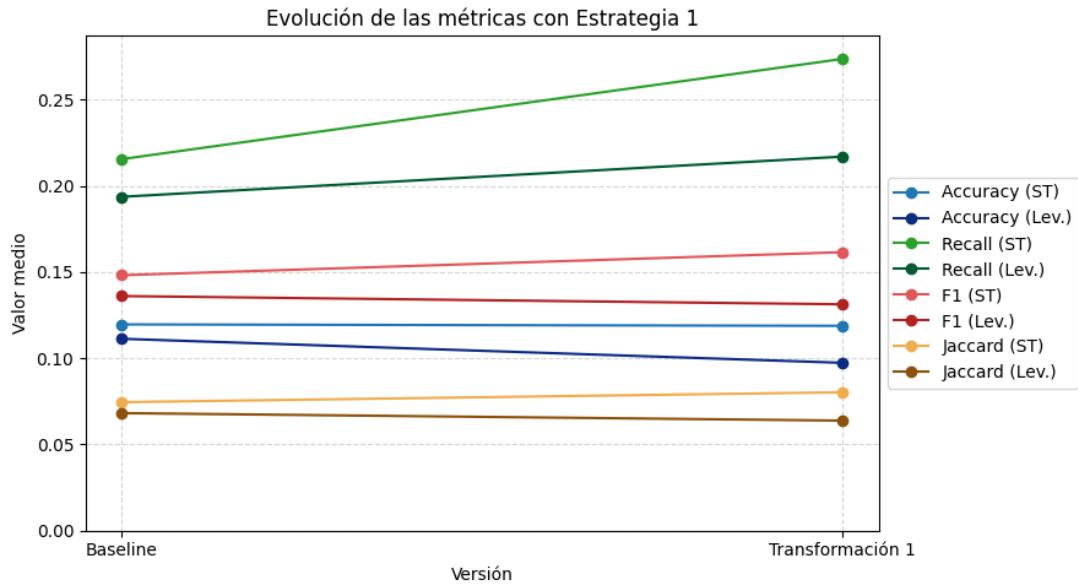


Figura 4.9: Evolución media de las métricas con Estrategia 1

En la figura puede apreciarse que, en promedio, el aumento de las coberturas supera la ligera disminución de las precisiones. Este comportamiento confirma que la estrategia logra elevar la cobertura de relaciones con un coste moderado en precisión. Además, la evolución de las métricas de Jaccard indica que la estructura global del grafo no se ve alterada de forma significativa.

#### 4.2.1.2. Estrategia 2: Extracción de Tripletas de Chunks con Verbos

La hipótesis de partida era que, al reutilizar los chunks verbales como entradas para el extractor de tripletas de OpenIE, se detectarían relaciones que habían quedado fuera en la extracción inicial, aumentando la cobertura sin añadir apenas ruido.

Al aplicar esta segunda estrategia, se generaron nuevas tripletas para cada uno de los textos de referencia, sin contar las ya presentes en el Baseline:

- *Utilitarianism*: 2 tripletas nuevas.
- *Ellipse*: 3 tripletas nuevas.
- *Homeostasis*: 5 tripletas nuevas.
- *Inflation*: 22 tripletas nuevas.

En este caso, al generar menos tripletas nuevas que con la estrategia 1, el impacto de la estrategia 2 será menor. Además, dado que algunas de estas tripletas coinciden con variantes ya presentes en el Baseline, es imprescindible evaluar su relevancia frente al grafo Gold utilizando las métricas correspondientes.

### Análisis por texto

En la gráfica del utilitarismo (Figura 4.10) se observa que las coberturas semántica y léxica permanecen constantes, mientras que las precisiones y los F1 scores disminuyen en torno al 14 % y al 6 % respectivamente. Esto sugiere que las dos tripletas adicionales generadas no aportan información relevante al grafo, sino que introducen ruido. Con solo dos tripletas nuevas respecto al Baseline, no hay una mejora en la calidad del grafo.

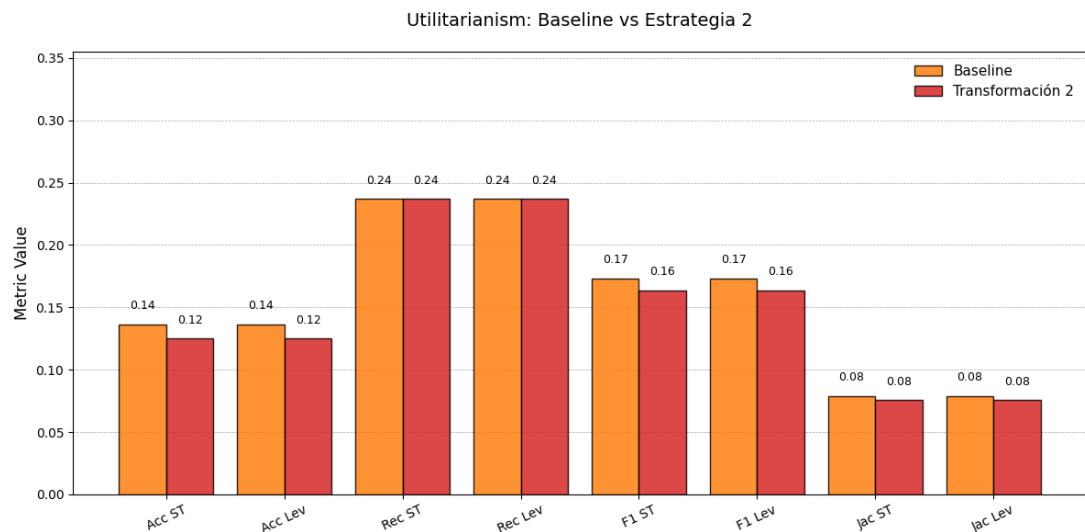


Figura 4.10: Utilitarianism: Baseline vs Estrategia 2

Para el texto de la Elipse ocurre algo similar (Figura 4.11). las coberturas semántica y léxica permanecen constantes, mientras que las precisiones y el resto de métricas disminuyen ligeramente. Esto sugiere que las tripletas extraídas de los chunks verbales tampoco aportan información relevante a este grafo.

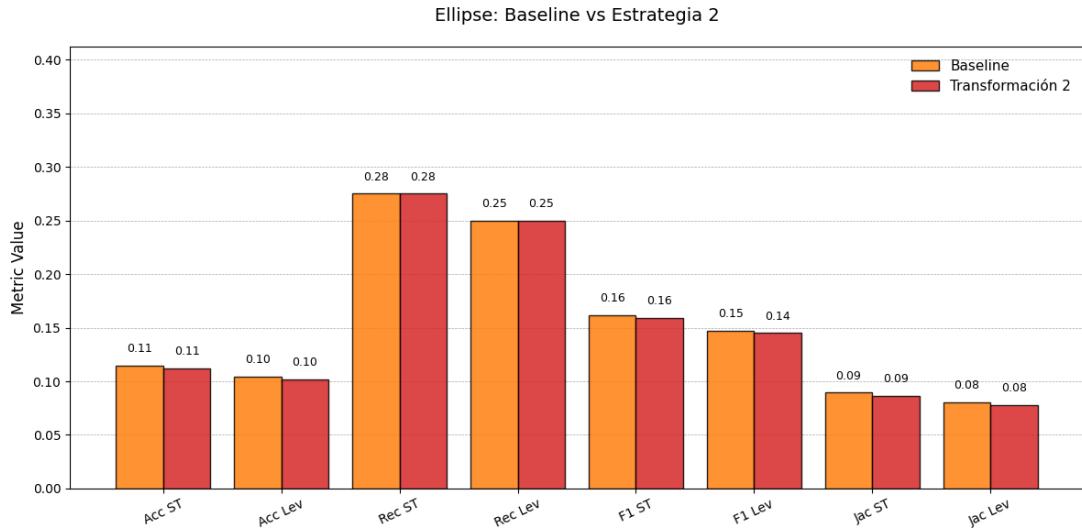


Figura 4.11: Ellipse: Baseline vs Estrategia 2

Para *Homeostasis*, se observa en la (Figura 4.12) que la estrategia resultó más beneficiosa. Todas las métricas presentan un pequeño incremento. Sin embargo, al haberse incorporado tan pocas tripletas al Baseline, el impacto global en las métricas tampoco es relevante.

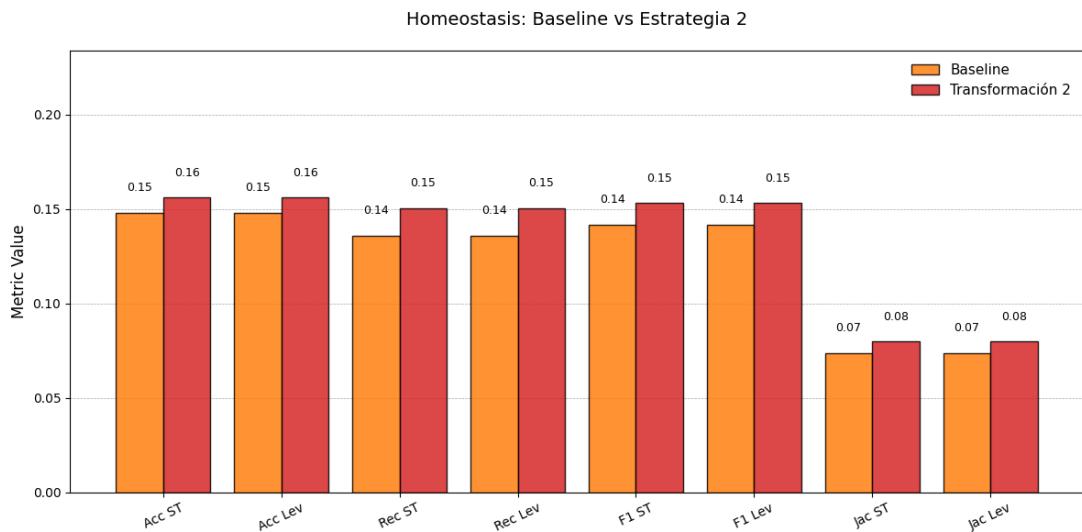


Figura 4.12: Homeostasis: Baseline vs Estrategia 2

Para *Inflation* (Figura 4.13) destaca la mejora de la cobertura semántica que crece en un 19 %. En este caso, las métricas semánticas mejoran más que las métricas léxicas, lo que indica que las tripletas añadidas con la estrategia 2 han aportado un valor semántico significativo al grafo Baseline del texto sobre inflación. Sin embargo, salvo la cobertura semántica, el resto de las métricas se mantienen prácticamente constantes.

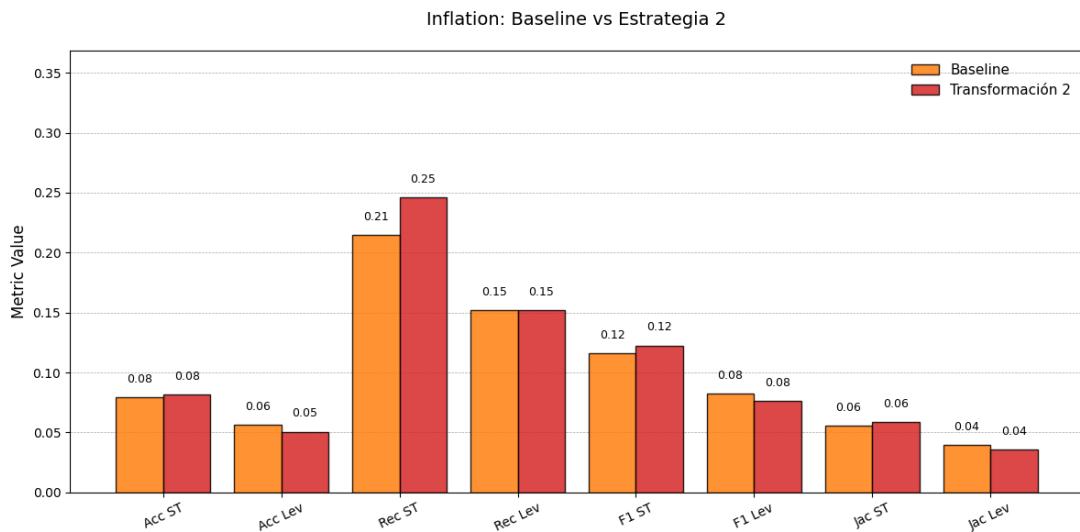


Figura 4.13: Inflation: Baseline vs Estrategia 2

En conjunto, estos resultados muestran que la estrategia 2 es poco consistente. En algunos textos, aporta un ligero aumento de cobertura, mientras que en otros casos, no solo carece de efecto, sino que incluso introduce ruido en el grafo. Además, las diferencias entre aplicarla o no aplicarla son siempre muy leves.

### Evolución media de las métricas

En la (Figura 4.14) se observa que la cobertura aumenta también con esta estrategia, aunque en menor medida que con la estrategia 1. Concretamente, la cobertura semántica mejora en un 5 % aproximadamente y la léxica en un 2 %. Por otro lado las precisiones empeoran pero también ligeramente. El F1 score es decisivo en este caso, mejorando un 1 % semánticamente y empeorando otro 1 % léxicamente. Ocurre lo mismo con la similitud estructural, el coeficiente de Jaccard semántico sube 1 %, pero el Jaccard léxico baja otro 1 %).

Estos resultados confirman que este método no constituye una mejora sólida ni generalizable, sino que su eficacia depende del tipo de texto analizado y que en cualquier caso, los efectos de aplicar la estrategia son siempre muy leves.

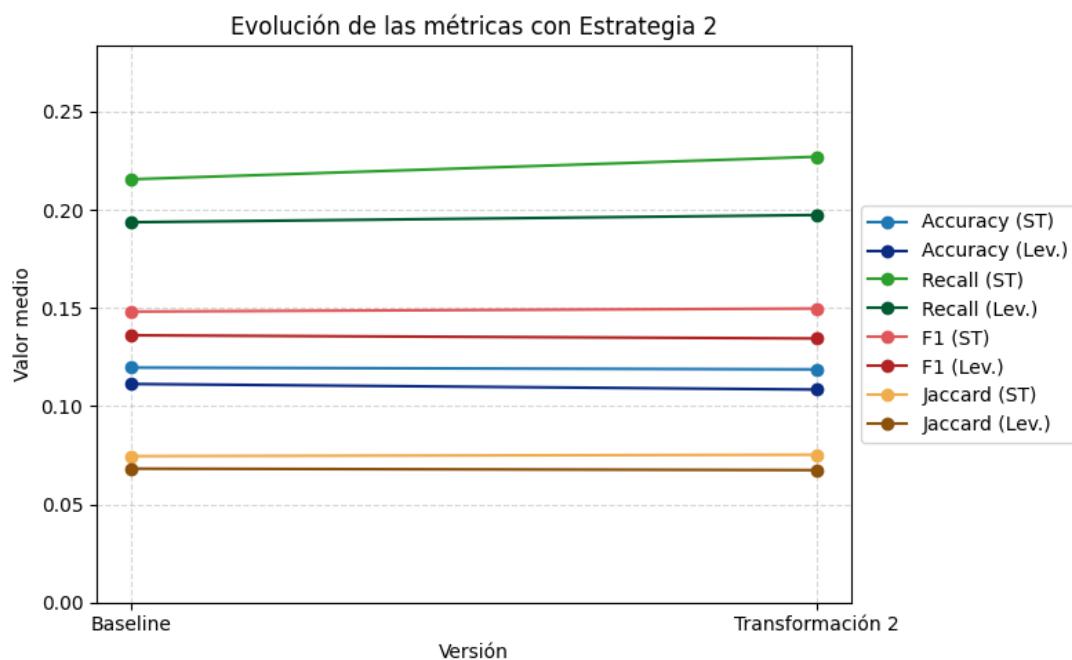


Figura 4.14: Evolución media de las métricas Estrategia 2

#### 4.2.1.3. Evaluación del Bloque I

Este apartado tiene como objetivo determinar cuáles de las estrategias evaluadas en este bloque deben aplicarse al grafo Baseline en las fases posteriores.

Por un lado, la Estrategia 1 (normalización mediante chunking y re-extracción) confirma de modo consistente la hipótesis inicial: en todos los casos, la re-extracción a partir de chunks unificados mejora la cobertura del grafo sin sacrificar de forma significativa ni la precisión ni la similitud global con el grafo Gold. Esto demuestra que convertir los chunks en tokens únicos y volver a aplicar OpenIE es un método eficaz para capturar relaciones completas que el extractor original había pasado por alto.

Por otro lado, la Estrategia 2 (extracción de triplets a partir de chunks verbales) solo aporta un ligero aumento de cobertura, muy variable según el texto, y en algunos casos incluso empeora las métricas más de lo que las mejora. Aunque en ocasiones añade relaciones nuevas y valiosas, su rendimiento no es sólido ni generalizable.

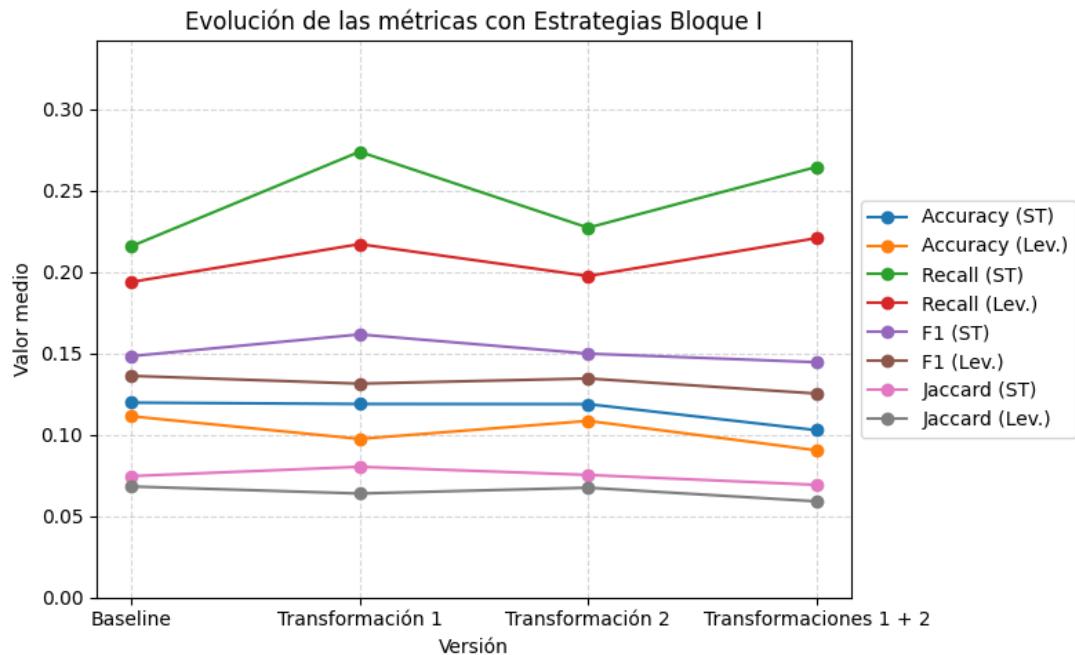


Figura 4.15: Evaluación Bloque I

La Figura 4.15, que compara las medias de las métricas del grafo Baseline, Transformación 1, Transformación 2 y la aplicación conjunta de ambas, muestra con claridad que la mejor opción es la Estrategia 1. Esta estrategia es la que más aumenta la cobertura de forma general sin afectar de manera significativa la precisión, lo que indica que no introduce ruido. Además, captura tripletas relevantes que la extracción inicial no había detectado. Por todo ello, se decide adoptar únicamente la Estrategia 1 como nueva base para las siguientes fases.

### 4.2.2. Bloque II: Homogeneización y Refinamiento

Este bloque aborda las limitaciones de OpenIE relacionadas con la delimitación incorrecta en las extracciones y el fenómeno de la extracción poco informativa, en el que las tripletas generadas omiten información clave. El objetivo de este conjunto de estrategias es refinar los predicados mal extraídos y unificar las entidades que hacen referencia a un mismo concepto.

#### 4.2.2.1. Estrategia 3: Corrección de Verbos Modales

La hipótesis inicial planteaba que, al incorporar los modales (“can”, “may”, “should”, “will”, etc.) en el propio predicado y limpiar al mismo tiempo los objetos, se obtendrían relaciones con predicados más informativos y objetos más “puros”, lo que debería aumentar la precisión del grafo.

Al aplicar esta transformación a los textos de *Utilitarianism* y *Ellipse*, los únicos en los que se habían detectado duplicaciones de verbos modales, se observa que predicados genéricos como “can” pasan a ser más completos al mismo tiempo que se limpian los objetos. Por ejemplo, “can describe by” sustituye al predicado genérico “can”, y el objeto “can describe by set of point in coordinate plane” se reduce a “set of point in coordinate plane”. De igual modo, “can discern throughout history of ethical theory” se separa en el predicado “can discern throughout” y el objeto “history of ethical theory”. En los textos *Homeostasis* e *Inflation* no hay tripletas con modales, por lo que sus métricas quedarán idénticas a las de la Transformación 1.

#### Análisis por texto

En el caso de *Utilitarianism*, como muestra la Figura 4.16, ambas precisiones experimentan una ligera mejora. La cobertura semántica permanece constante, mientras que la cobertura léxica destaca por aumentar un 10%. Este incremento probablemente se deba al proceso de limpieza de predicados y objetos. En conjunto, el aumento en estas métricas refleja una mejora en la calidad del grafo.

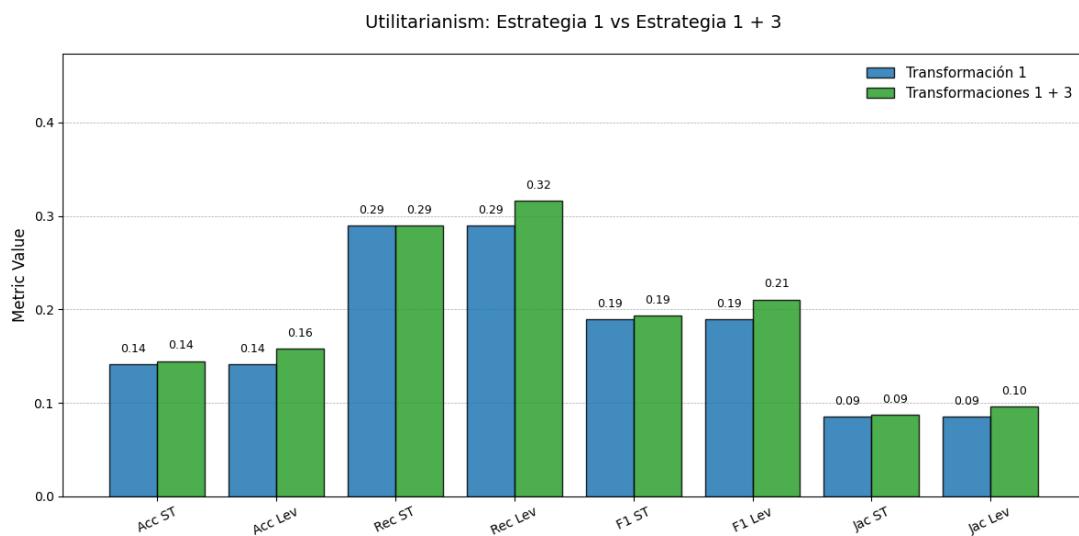


Figura 4.16: Utilitarianism: Estrategia 1 vs Estrategia 1 + 3

En el caso de la *Ellipse* (Figura 4.17), las mejoras son aún más destacables. Las precisiones semántica y

léxica mejoran un 15 % y un 33 % respectivamente. Además, la limpieza de verbos y objetos parece haber mejorado también las coberturas en este caso. El aumento de las métricas de Jaccard, refleja también una mejora clara en la estructura del grafo

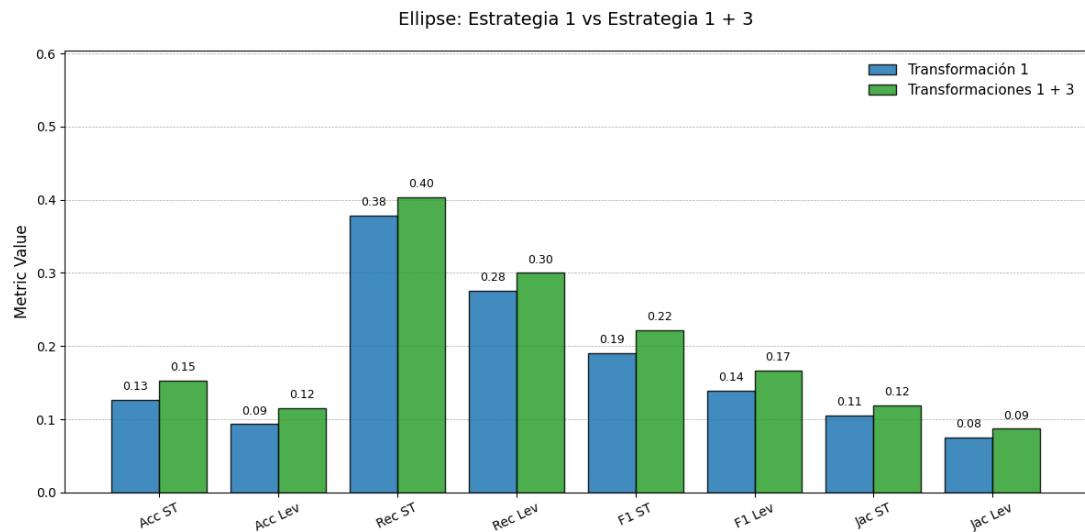


Figura 4.17: Ellipse: Estrategia 1 vs Estrategia 1 + 3

En conjunto, la corrección de los verbos modales ha limpiado de manera efectiva las tripletas afectadas en los textos del Utilitarismo y la elipse, lo que se traduce en un aumento notable tanto de la precisión como de la cobertura del grafo.

### Evolución media de las métricas

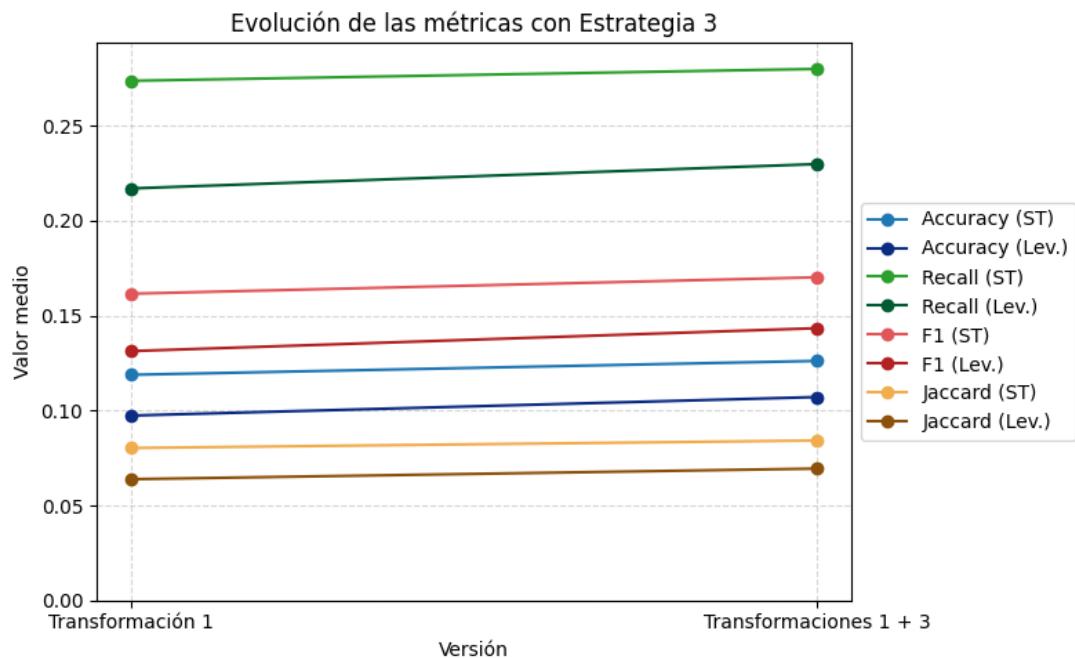


Figura 4.18: Evolución media de las métricas Estrategia 3

En la media de las métricas de *Utilitarianism* y *Ellipse* (Figura 4.17) se aprecia con claridad que la Transformación 3 mejora todas las métricas. Esto confirma la hipótesis de que, al depurar los verbos modales y limpiar los objetos, se obtienen predicados más completos y objetos más precisos, lo que incrementa la precisión del grafo y su calidad general.

#### 4.2.2.2. Estrategia 4: Mapeo de Sujetos y Objetos a Chunks de CoreNLP

La hipótesis de la Estrategia 4 plantea que, al reemplazar en el grafo generado por OpenIE las entidades originales por sintagmas nominales más limpios y gramaticalmente correctos, se obtendría un grafo más limpio y más preciso.

##### Análisis por texto

En el caso del texto del Utilitarismo, se puede observar en la (Figura 4.19) que la aplicación de la Estrategia 4 sí mejora los resultados de forma general. Las métricas semánticas experimentan un aumento más notable, mientras que las métricas léxicas se mantienen prácticamente constantes. Lo más destacado es el incremento en la cobertura semántica, posiblemente debido a que el mapeo de chunks ha permitido que aparezcan en el grafo sintagmas más completos que antes no se capturaban.

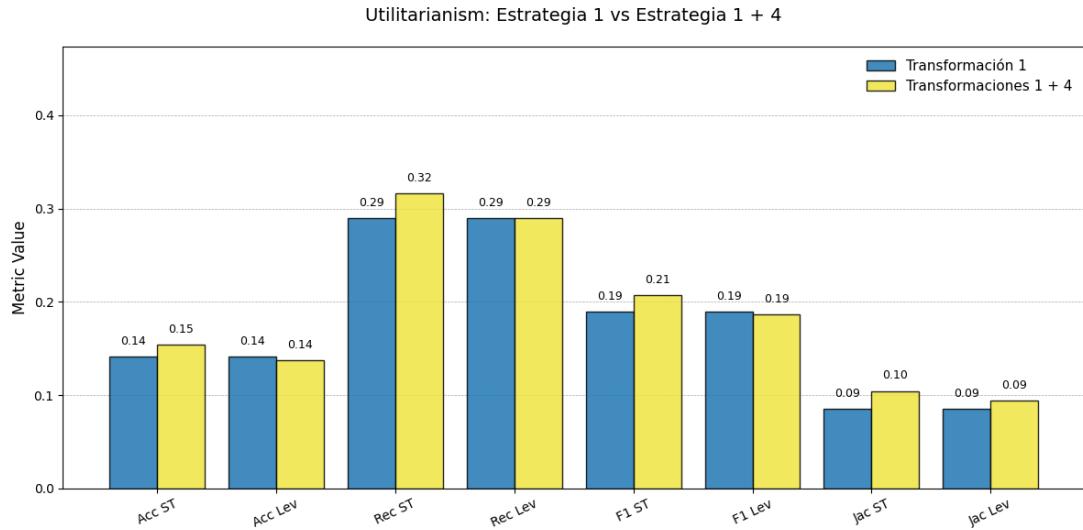


Figura 4.19: Utilitarianism: Estrategia 1 vs Estrategia 1 + 4

En el caso de *Ellipse* (Figura 4.20), los resultados empeoran. Todas las precisiones y coberturas caen de forma notable respecto al grafo con la Transformación 1, especialmente las métricas léxicas. Esto indica que los chunks extraídos por CoreNLP y utilizados para el mapeo son menos precisos que las entidades identificadas originalmente por OpenIE.

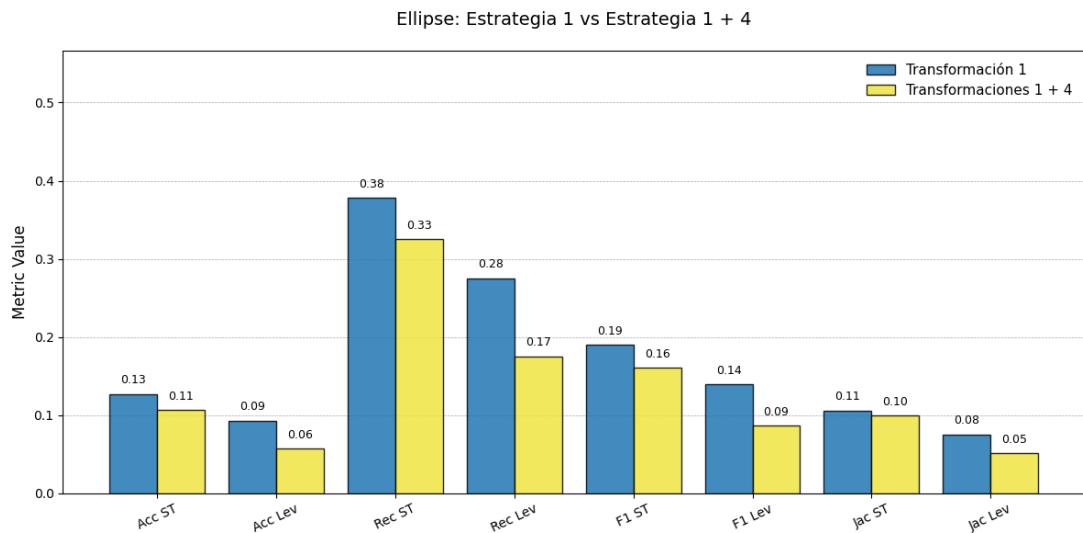


Figura 4.20: Ellipse: Estrategia 1 vs Estrategia 1 + 4

Para el texto de la Homeostasis, en la Figura 4.21 se aprecia una clara diferencia entre las métricas semánticas y las léxicas. Por ejemplo, mientras que la precisión semántica mejora en un 7%, la léxica empeora un 60%. Esto indica que, aunque las entidades generadas ahora están más alejadas en superficie del grafo Gold, en realidad se acercan más a nivel de significado. Dicho de otro modo, los chunks extraídos

por CoreNLP, a pesar de a escribirse de forma distinta, comparten un sentido más parecido con los nodos del grafo Gold.

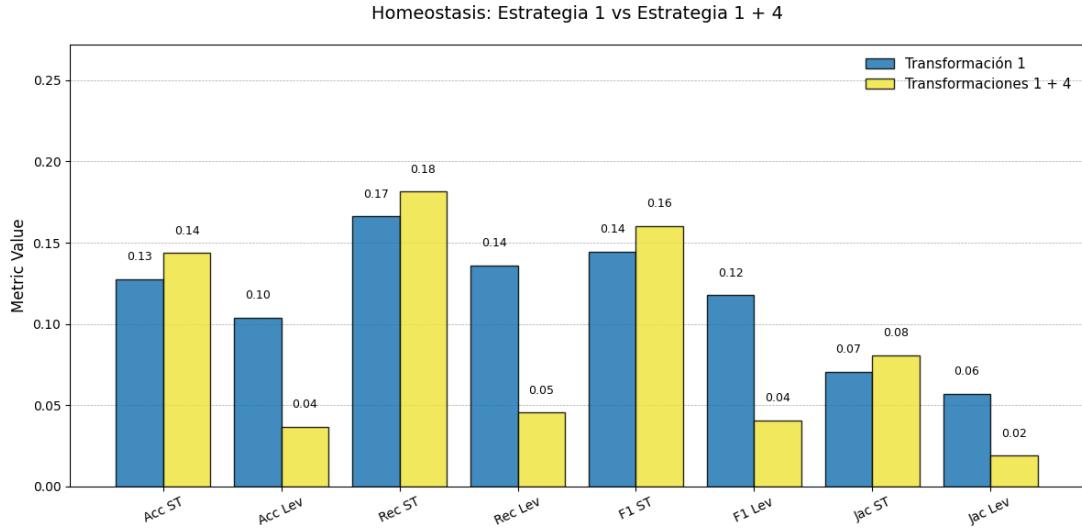


Figura 4.21: Homeostasis: Estrategia 1 vs Estrategia 1 + 4

En el caso de la inflación (Figura 4.22) se aprecian mejoras, aunque bastante leves. La única métrica que disminuye es la cobertura léxica, lo que indica que los nuevos nodos están algo más alejados en su forma escrita del grafo Gold. Sin embargo, en términos generales, la Estrategia 4 ha aportado una ligera mejora global para el texto sobre la inflación.

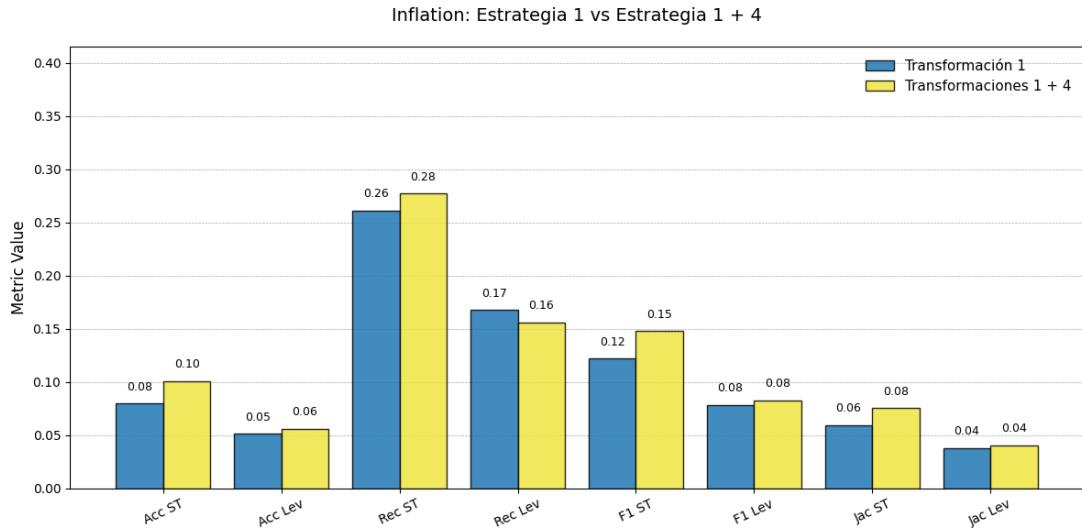


Figura 4.22: Inflation: Estrategia 1 vs Estrategia 1 + 4

En conjunto, aunque la Estrategia 4 muestra potencial, en algunos casos como el de elipse, los resultados empeoran. Esto sugiere que sería necesario mejorar la calidad de los chunks utilizados en

el mapeo, para que representen de forma más fiel las entidades de cada frase y permitan mapear sujetos y objetos con mayor precisión.

### Evolución media de las métricas

En la (Figura 4.23) se aprecia que únicamente las métricas semánticas mejoran al aplicar la Estrategia 4. La precisión semántica experimenta un incremento del 6 %, mientras que la precisión léxica empeora un 26 %. De igual modo, la cobertura semántica sube menos de un 1 %, pero la cobertura léxica desciende un 23 %. Los F1 scores también reflejan esto. En cuanto a la similitud estructural, el coeficiente de Jaccard semántico aumenta un 12 %, mientras que el Jaccard léxico disminuye un 20 %.

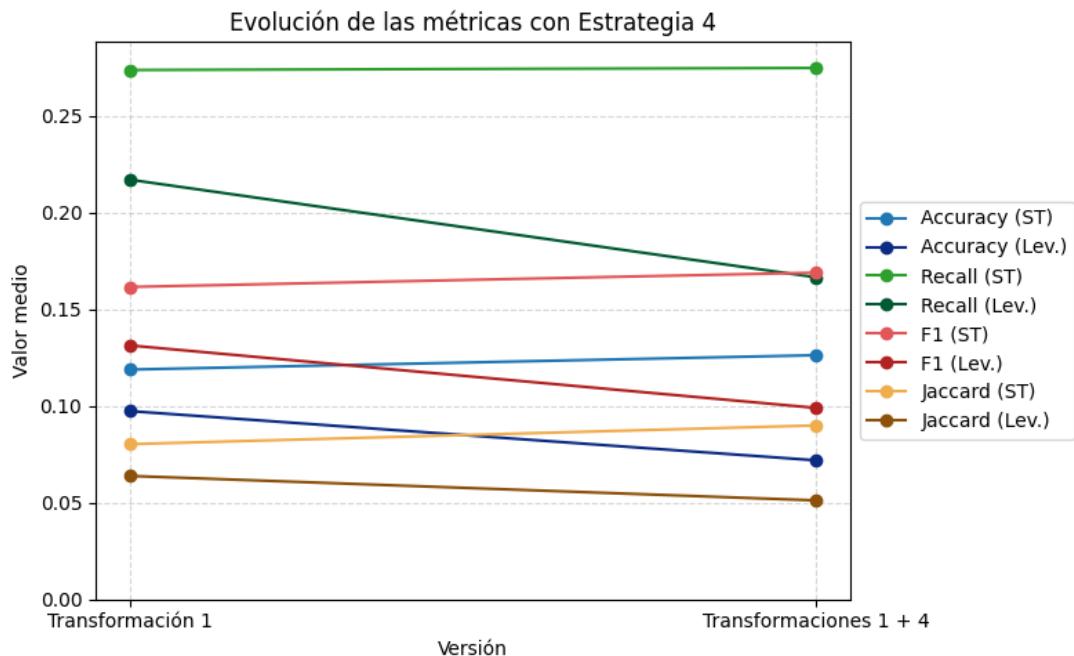


Figura 4.23: Evolución media de las métricas Estrategia 4

Los resultados indican que, aunque la estrategia depura el grafo desde el punto de vista semántico, no mejora su calidad léxica. Por tanto, convendría optimizar tanto la extracción de chunks con CoreNLP como el propio proceso de mapeo, de modo que los sujetos y objetos resultantes reflejen con mayor fidelidad las entidades originales del texto.

#### 4.2.2.3. Estrategia 5: Mapeo de Sujetos y Objetos con Chunks Limpiados y Enriquecidos (CoreNLP + SpaCy)

La hipótesis señalaba que, al mejorar y enriquecer la calidad de los chunks y al flexibilizar el proceso de mapeo, se aumentaría el número de asignaciones correctas de sujetos y objetos, traduciéndose en un grafo más limpio y preciso.

La Estrategia 5 retoma la Estrategia 4 pero incorpora mejoras en la calidad de los chunks y flexibiliza el mapeo para optimizar los resultados. En primer lugar, se eliminan las stop words iniciales de cada chunk. Después, se descartan aquellos chunks que contengan verbos para preservar únicamente sintagmas

nominales. Finalmente, se enriquecen los candidatos añadiendo los sintagmas nominales extraídos por SpaCy.

### Análisis por texto

En el caso de *Utilitarianism* (Figura 4.20), se aprecia una mejora en todas las métricas. Al compararlo con los avances que ofrecía la estrategia anterior en este texto del dataset de referencia, se observa que ahora la mejora es más notable. Especialmente en el ámbito léxico, donde con la estrategia previa las métricas se mantenían relativamente constantes frente al grafo de la Transformación 1.

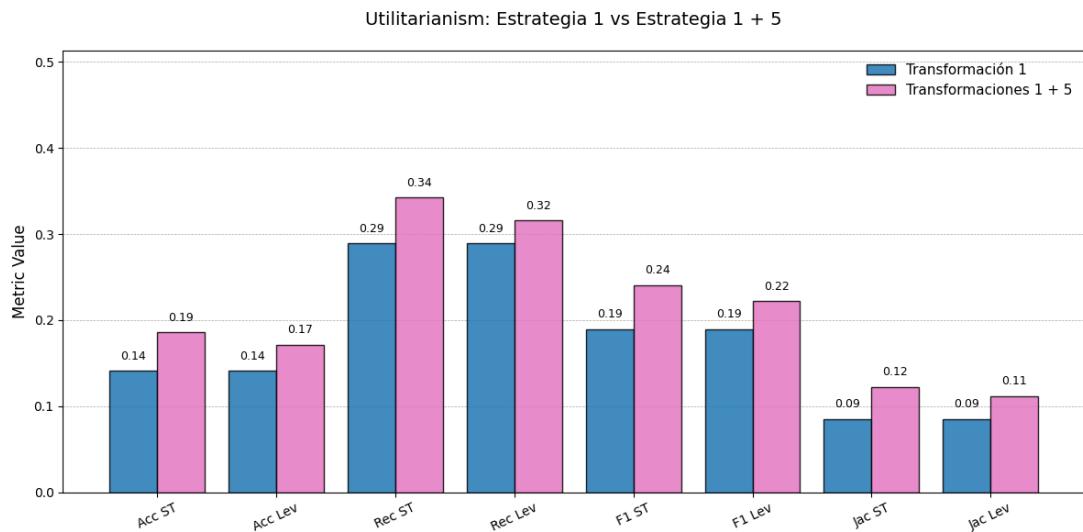


Figura 4.24: Utilitarianism: Estrategia 1 vs Estrategia 1 + 5

Para el texto de la elipse (Figura 4.20), mientras que con el mapeo de la estrategia anterior casi todas las métricas empeoraban, ahora con los nuevos chunks y el mapeo flexible se puede observar una gran mejora. La cobertura léxica es la única métrica que no mejora respecto al grafo obtenido en el bloque anterior. Aun así, su descenso es menor que el experimentado con la Estrategia 4. Esto sugiere que los nuevos chunks limpios y enriquecidos son más cercanos a los del grafo Gold, y que además están llevando a más entidades a ser mapeadas a chunks precisos y coherentes.

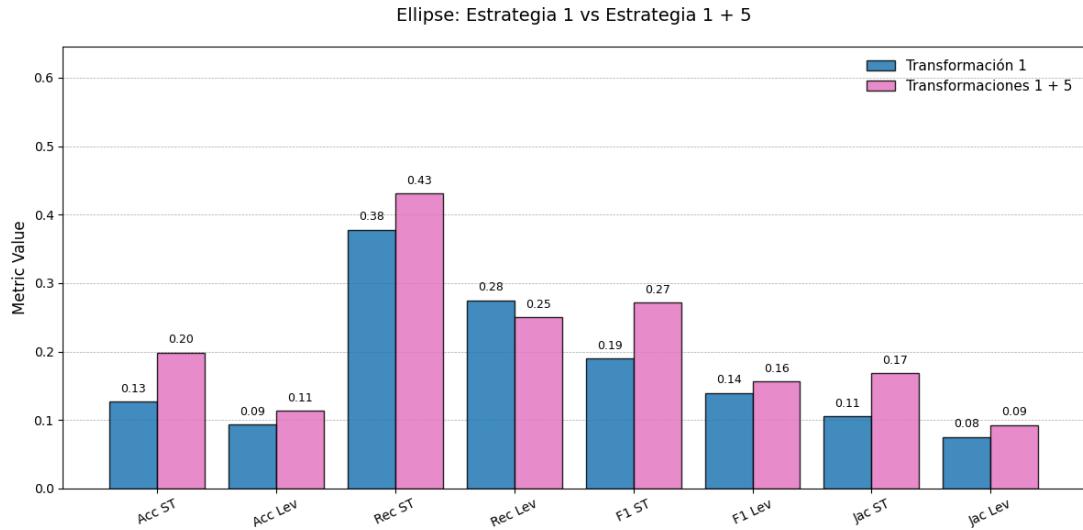


Figura 4.25: Ellipse: Estrategia 1 vs Estrategia 1 + 5

Para el tercer texto del dataset de referencia, el de la homeostasis, se aprecia en la Figura 4.21 de nuevo una mejora general, sobre todo a nivel semántico, donde los avances son más pronunciados. A diferencia de la Estrategia 4 en la que todas las métricas léxicas empeoraban de forma notable, con la nueva estrategia solo disminuye significativamente la cobertura léxica, y lo hace en una magnitud mucho menor (con la Estrategia 4 empeoraba en un 66%, mientras que ahora solo lo hace un 22%).

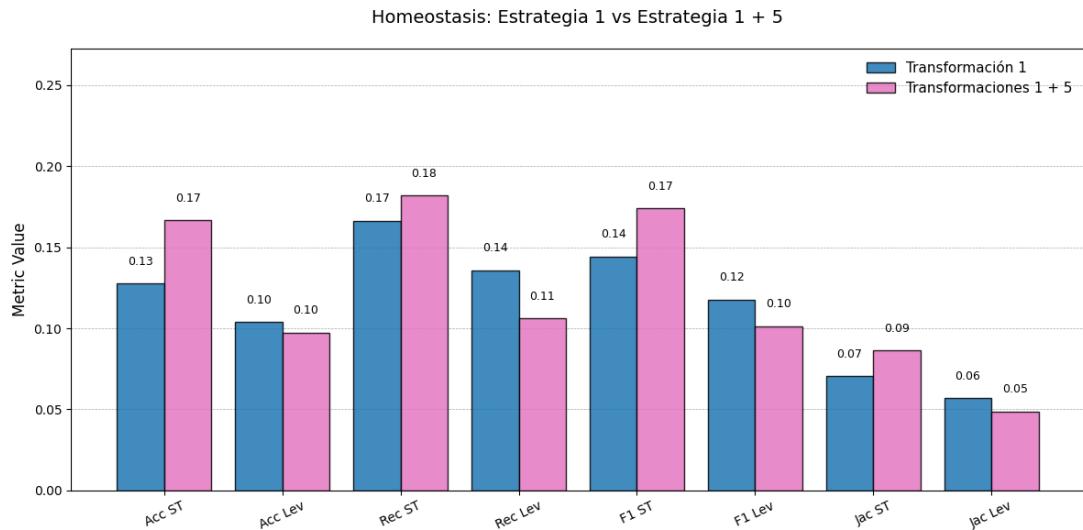


Figura 4.26: Homeostasis: Estrategia 1 vs Estrategia 1 + 5

Nuevamente, para *Inflation* (Figura 4.22) mientras con la Estrategia 4 las métricas se mantenían casi constantes, ahora se aprecia una mejora destacada, con un incremento muy destacable en las precisiones. Las coberturas apenas varían, lo que sugiere que la extracción de chunks previa ya capturaba adecuadamente las entidades del texto. Los F1 scores reflejan ese aumento notable en las precisiones y la

similitud estructural con el grafo Gold también mejora, el coeficiente de Jaccard semántico mejora en un 66 % y el léxico mejora en un 50 %.

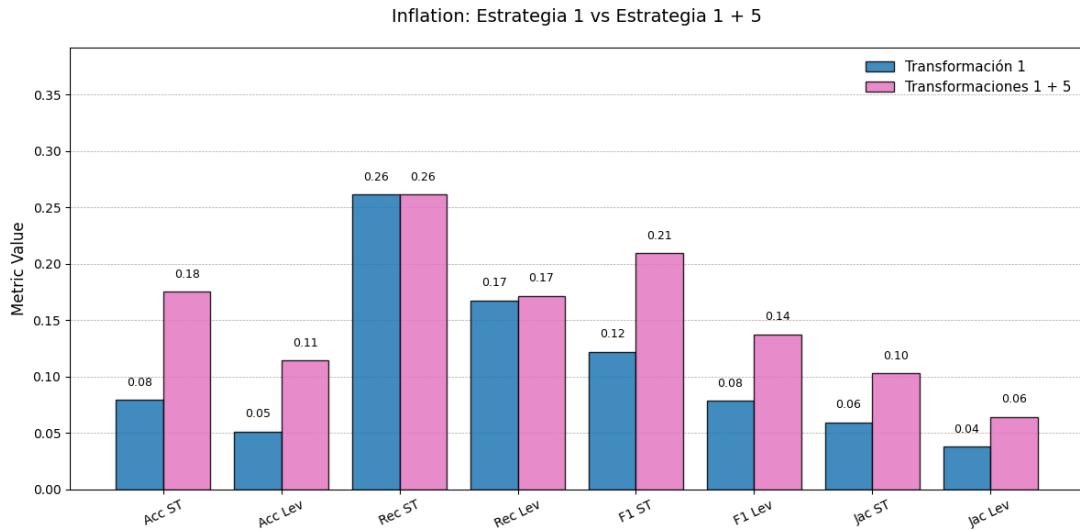


Figura 4.27: Inflation: Estrategia 1 vs Estrategia 1 + 5

En este análisis por texto se evidencia de nuevo que mapear entidades a chunks no solo mejora la precisión del grafo, sino que también amplía su cobertura. Esto puede deberse a que muchas de las entidades extraídas por OpenIE son menos precisas que emplear directamente los sintagmas nominales detectados en la frase.

También se observa que, aunque esta estrategia puede disminuir ligeramente la cobertura, sobretodo léxica en algunos casos, la mejora sustancial en el resto de las métricas compensa con creces. Esta pequeña caída en la cobertura puede explicarse porque ahora existen menos formas de referirse a un mismo concepto: dos expresiones que antes se representaban con ligeras variaciones léxicas se unifican en un solo mapeo. Es posible que, al hacerlo, la entidad aparezca con una forma menos parecida en superficie al nodo correspondiente del grafo Gold, pero al mismo tiempo se evita tener dos nodos que representen la misma entidad, lo cual mejora la consistencia y claridad global del KG.

### Evolución media de las métricas

En la Figura 4.28 se aprecia que la Estrategia 5 mejora de manera global la calidad del grafo, superando con creces los resultados obtenidos con la Estrategia 4.

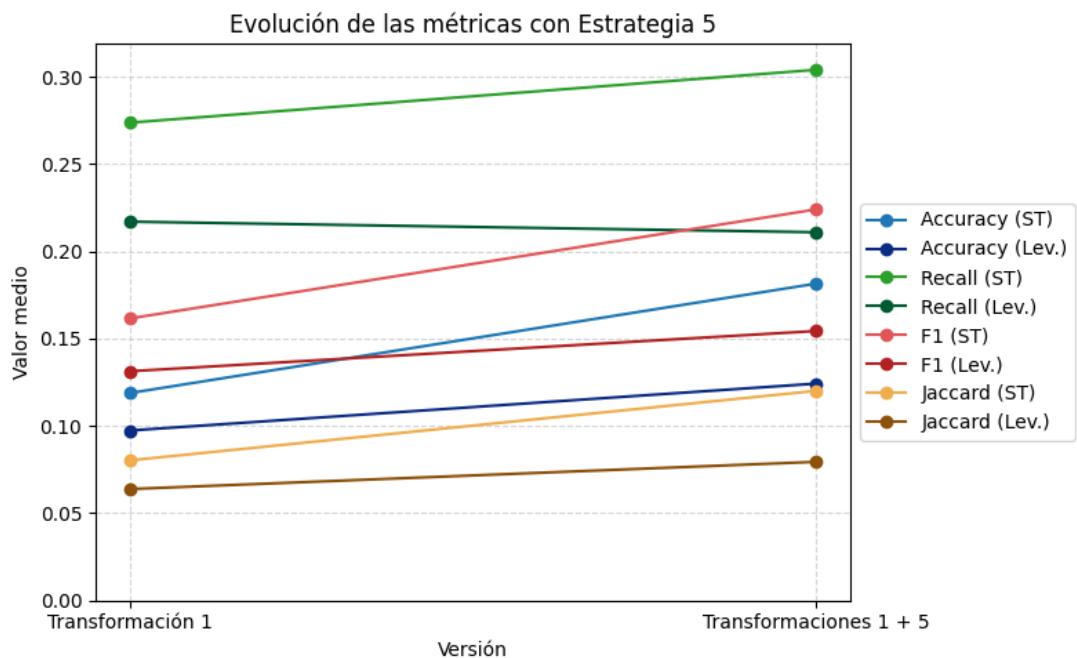


Figura 4.28: Evolución media de las métricas Estrategia 5

Todas las métricas experimentan mejoras de entre un 10 % y un 35 %, salvo la cobertura léxica, que se reduce en menos de un 1 %. Como se explicó anteriormente, esta ligera caída puede deberse a la desaparición de algunas variantes léxicas de las entidades a causa del mapeo. En conjunto, estos resultados confirman que, al mejorar y enriquecer la calidad de los chunks y al flexibilizar el proceso de mapeo, se obtiene un grafo más limpio y preciso, sobre todo a nivel semántico.

#### 4.2.2.4. Evaluación del Bloque II

El objetivo de este bloque era unificar entidades equivalentes y refinar predicados mal extraídos, garantizando una representación más coherente y clara de las relaciones.

En primer lugar, los resultados de la Estrategia 3, diseñada para corregir los verbos modales que provocan extracciones erróneas de tripletes por parte de OpenIE, confirman que esta intervención incrementa la precisión y la calidad general del grafo.

Esto indica que la Estrategia 3 será beneficiosa en los casos en que los verbos modales dificulten la extracción de triplets coherentes, cumpliendo así parte del objetivo de este segundo bloque de estrategias. Por tanto, se decide implementarla de aquí en adelante.

En segundo lugar, tras aplicar la Estrategia 3, disponemos de dos enfoques para el mapeo a chunks: un método más estricto que utiliza directamente los chunks de CoreNLP (Estrategia 4) o un enfoque más flexible que emplea los chunks previamente limpiados y enriquecidos con los sintagmas nominales extraídos por spaCy.

En la Figura 4.29 puede verse que la Estrategia 5 ofrece la mayor mejora global de resultados, a cambio de una ligera reducción de la cobertura léxica respecto al grafo obtenido tras aplicar las mejoras del Bloque I.

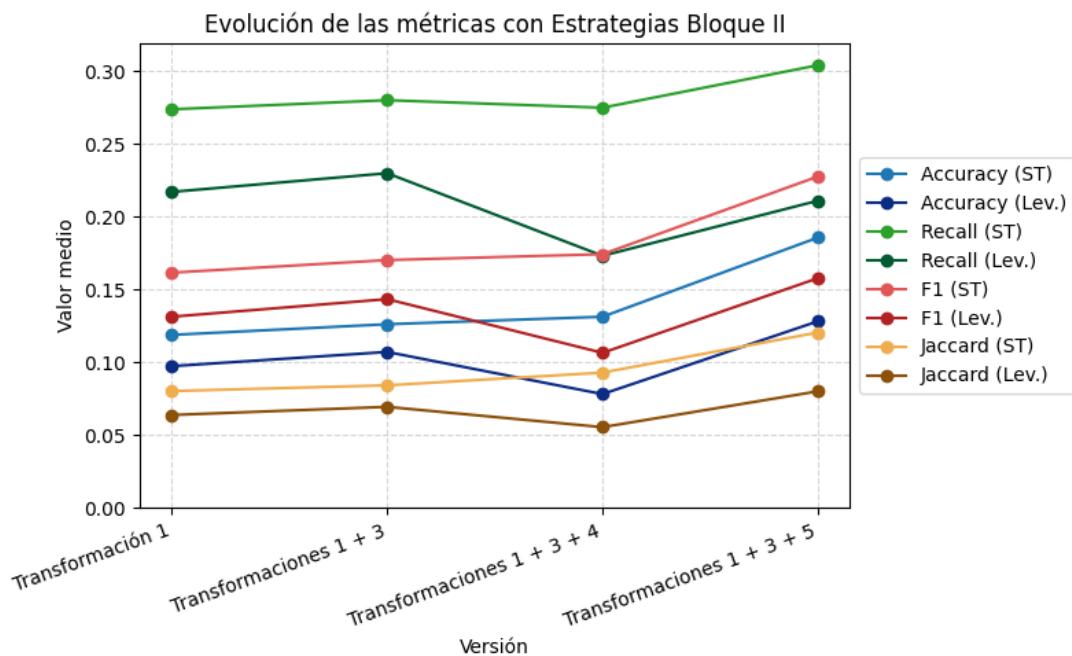


Figura 4.29: Evaluación Bloque II

Por tanto, en este bloque se optará por las Estrategias 3 y 5, ya que son las que mejor cumplen el objetivo de unificar entidades equivalentes y depurar predicados mal extraídos, logrando así una mejora significativa de la calidad general del grafo.

### 4.2.3. Bloque III: Mejora de Precisión

En este último bloque, se aborda el problema señalado en el estado del arte de los sistemas OpenIE relacionado con la extracción redundante de triplets. El objetivo es eliminar aquellas triplets redundantes o de baja confianza, con el fin de incrementar la precisión y reducir el ruido estructural.

#### 4.2.3.1. Estrategia 6: Filtrado de Triplets por Confianza

La hipótesis de partida para esta estrategia era que las triplets con puntuaciones de confianza bajas suelen corresponder a relaciones ambiguas, incorrectas o mal formadas. Su eliminación incrementaría la precisión global del grafo sin comprometer sustancialmente la cobertura.

Dado que el porcentaje de triplets con una confianza inferior a 1 en el conjunto de referencia es muy bajo, se espera que el impacto de esta estrategia sea reducido. Es necesario analizar las métricas para confirmarlo.

#### Análisis por texto

En el caso del Utilitarismo, como muestra la Figura 4.30, la eliminación de las triplets con nivel de confianza inferior a 1 no ha supuesto ninguna mejora en las métricas del grafo. Esto indica que esas triplets, además de ser precisas, contenían conceptos relevantes que también aparecen en el grafo Gold.

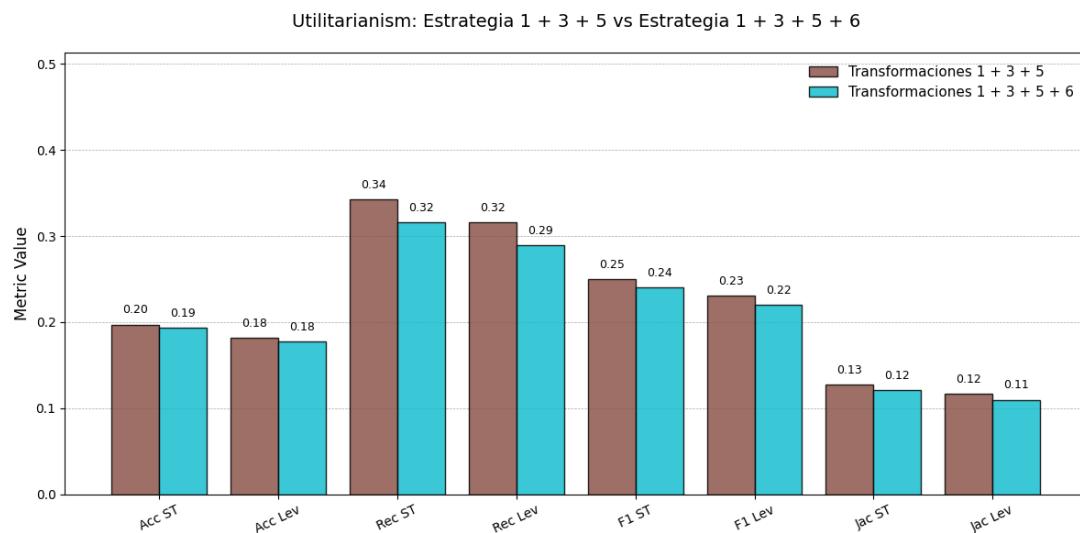


Figura 4.30: Utilitarianism: Estrategia 1 + 3 + 5 vs Estrategia 1 + 3 + 5 + 6

En el caso de la Elipse (Figura 4.31) sucede lo mismo que en el texto del Utilitarismo. Las coberturas semántica y léxica empeoran especialmente entre un 20% y un 35%, lo que indica que las triplets eliminadas contenían entidades y relaciones relevantes que sí aparecen en el grafo Gold. Aunque la precisión semántica mejora ligeramente, esta ganancia no compensa la pérdida en cobertura, reflejándose esto en el F1 score.

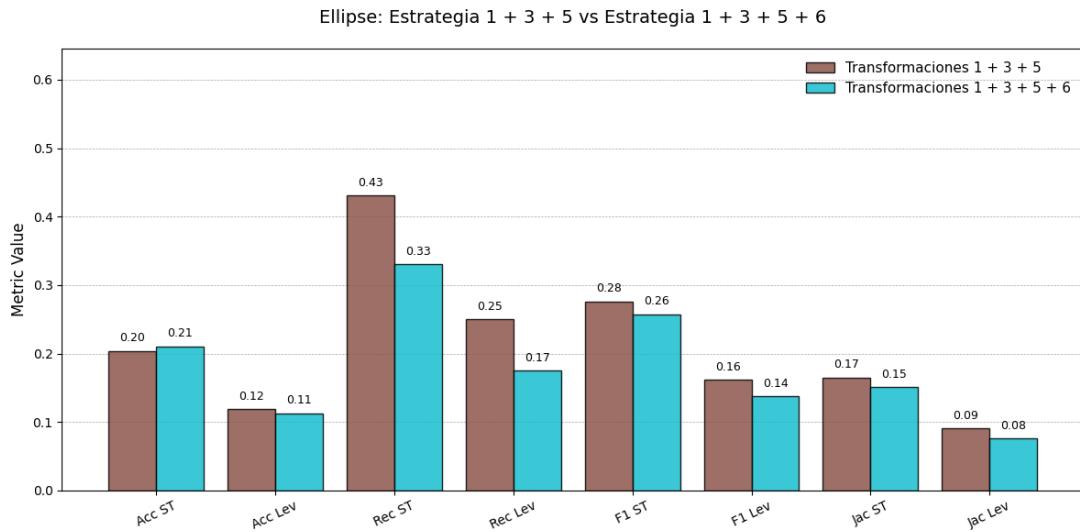


Figura 4.31: Ellipse: Estrategia 1 + 3 + 5 vs Estrategia 1 + 3 + 5 + 6

Para *Homeostasis* (Figura 4.32) casi todas las métricas se mantienen constantes, a excepción de las precisiones, que mejoran muy ligeramente. Esto puede explicarse porque la información contenida en las tripletas eliminadas también está presente en otras tripletas. Al retirarlas, la precisión aumenta un poco, pero la calidad general del grafo permanece prácticamente igual.

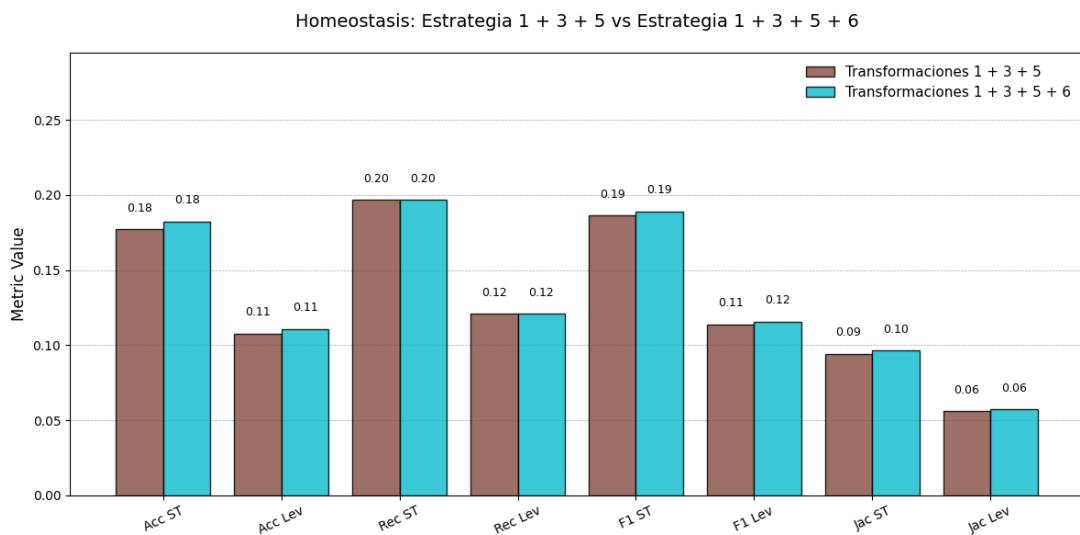


Figura 4.32: Homeostasis: Estrategia 1 + 3 + 5 vs Estrategia 1 + 3 + 5 + 6

Para el texto sobre la inflación se observa en la Figura 4.32 que la Estrategia 6 consigue una ligera mejora en la precisión léxica, lo que sugiere que elimina variantes de expresión alejadas del grafo Gold. Sin embargo, el resto de las métricas o bien se mantienen constantes o empeoran.

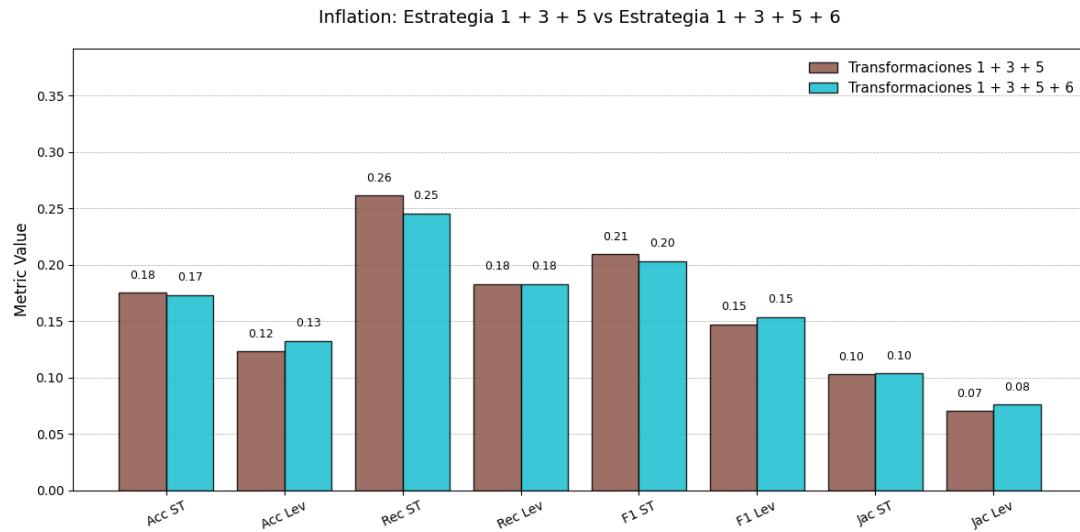


Figura 4.33: Inflation: Estrategia 1 + 3 + 5 vs Estrategia 1 + 3 + 5 + 6

En conjunto, el filtrado de tripletas por nivel de confianza no ejerce un impacto significativo en los textos de referencia. Las escasas mejoras que aporta resultan poco consistentes.

#### Evolución media de las métricas

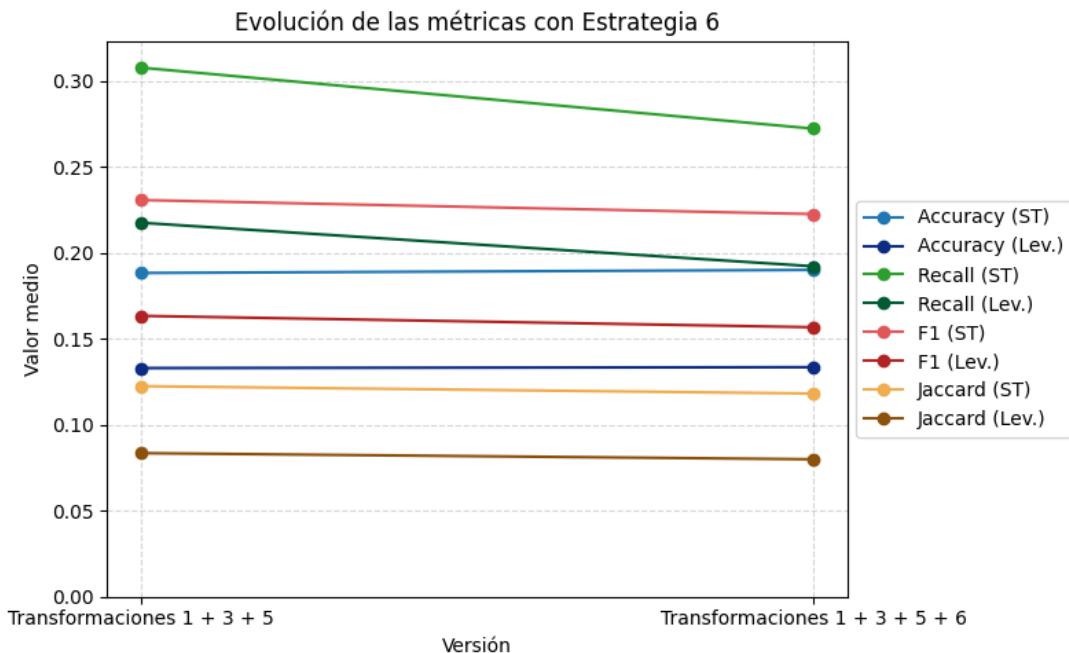


Figura 4.34: Evolución media de las métricas Estrategia 6

La Figura 4.34 confirma lo anterior: la precisión semántica y léxica apenas suben un 1%, mientras que

el resto de métricas retroceden, lo que indica que los pequeños avances en precisión no logran equilibrar las pérdidas en cobertura y coherencia estructural.

#### 4.2.3.2. Estrategia 7: Eliminación de Triplets Similares

La hipótesis de la Estrategia 7 plantea que depurar las tripletas redundantes incrementará la precisión del grafo al evitar la sobre-representación de las mismas relaciones. Para ello se sigue el proceso descrito en la metodología, que no solo elimina las redundancias, sino que también enriquece relaciones para hacerlas más completas.

##### Análisis por texto

En el caso del Utilitarismo, la aplicación de la Estrategia 7 produce un incremento notable de la precisión. Como se observa en la Figura 4.35 la precisión semántica sube en un 20% y la léxica un 22%. Asimismo, la similitud estructural con el grafo Gold mejora en un 38% a nivel semántico y en un 33% a nivel léxico. Que la cobertura permanezca constante indica que no se han eliminado nodos ni arcos relevantes.

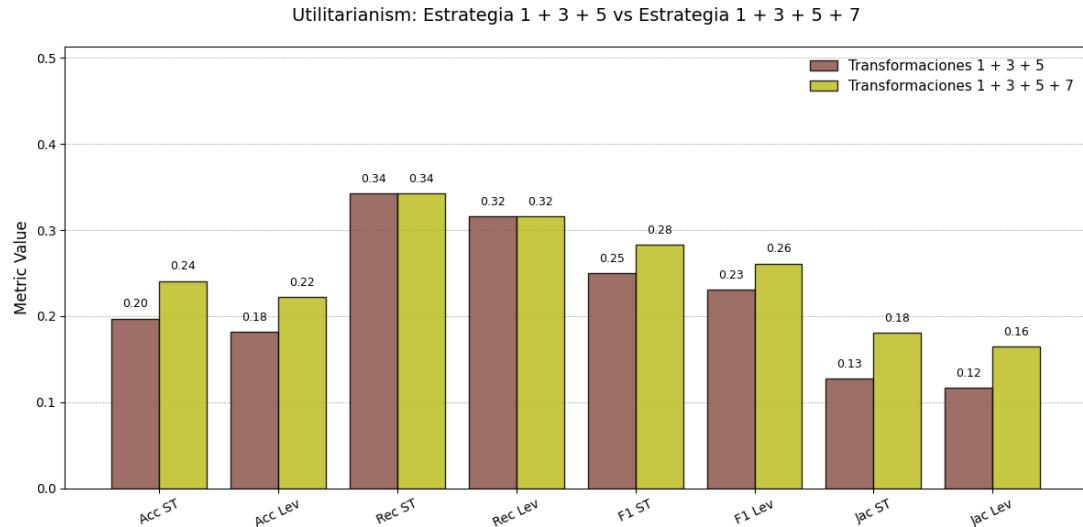


Figura 4.35: Utilitarianism: Estrategia 1 + 3 + 5 vs Estrategia 1 + 3 + 5 + 7

En el caso de Elipse (Figura 4.36), aunque aumentan tanto las precisiones como los coeficientes de Jaccard, se aprecia un claro empeoramiento de la cobertura semántica. Esto indica que, durante la depuración de tripletas, se han eliminado algunas relaciones que aportaban un significado importante y que sí estaban presentes en el grafo Gold. Esro hace el F1 score semántico también empeore ligeramente.

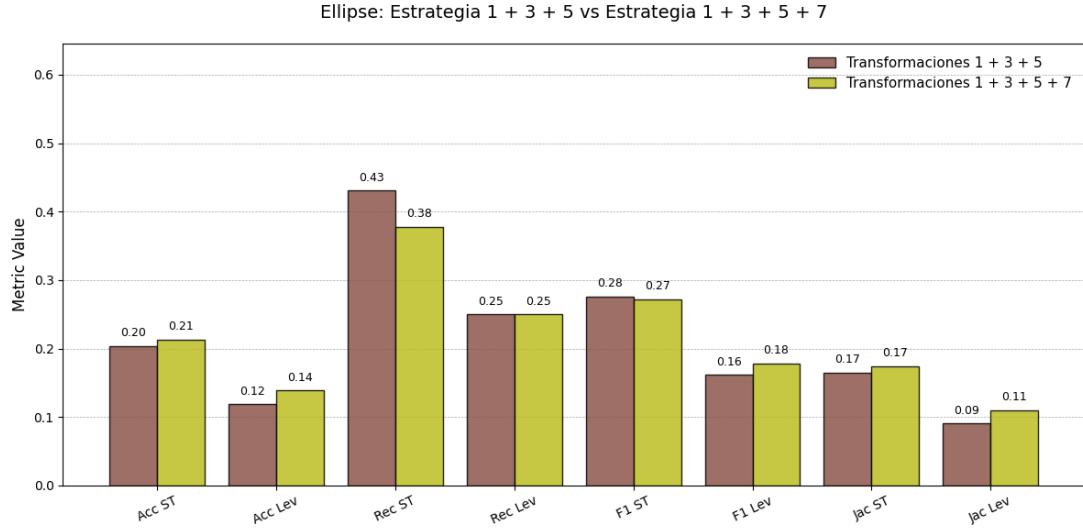


Figura 4.36: Ellipse: Estrategia 1 + 3 + 5 vs Estrategia 1 + 3 + 5 + 7

En *Homeostasis* (Figura 4.37) ocurre lo mismo que en *Utilitarianism*. Las coberturas (tanto semántica como léxica) permanecen constantes, mientras que las precisiones y las similitudes estructurales mejoran. Destaca una mejora del 28 % en la precisión semántica, lo que indica que se han eliminado muchas tripletas redundantes sin perder información relevante.

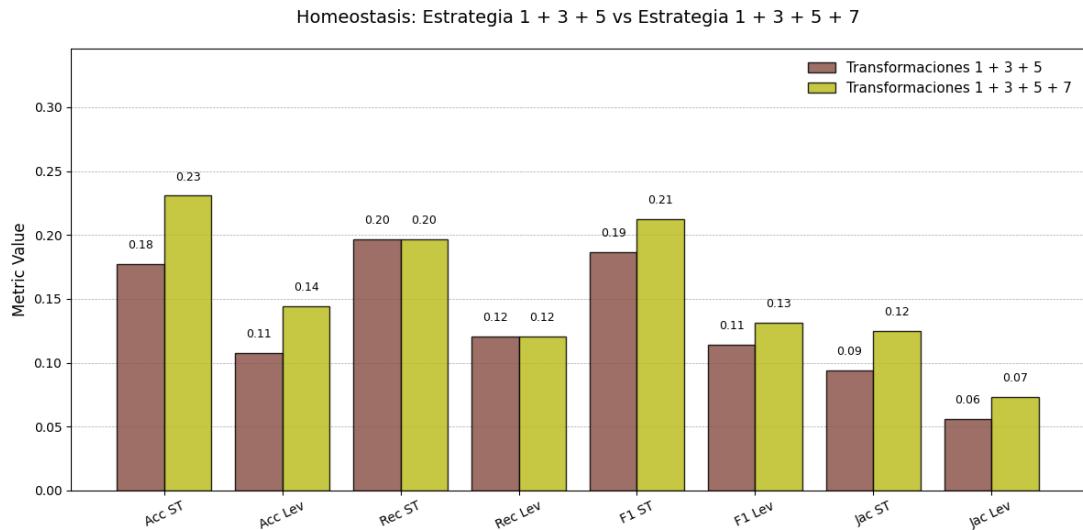


Figura 4.37: Homeostasis: Estrategia 1 + 3 + 5 vs Estrategia 1 + 3 + 5 + 7

En el caso de *Inflation* (Figura 4.38), las coberturas semántica y léxica disminuyen ligeramente, pero la ganancia en precisión es más notable, sobretodo en la precisión semántica. Esto indica que, a pesar de perder algo de amplitud en la captura de relaciones, las tripletas retenidas son más exactas.

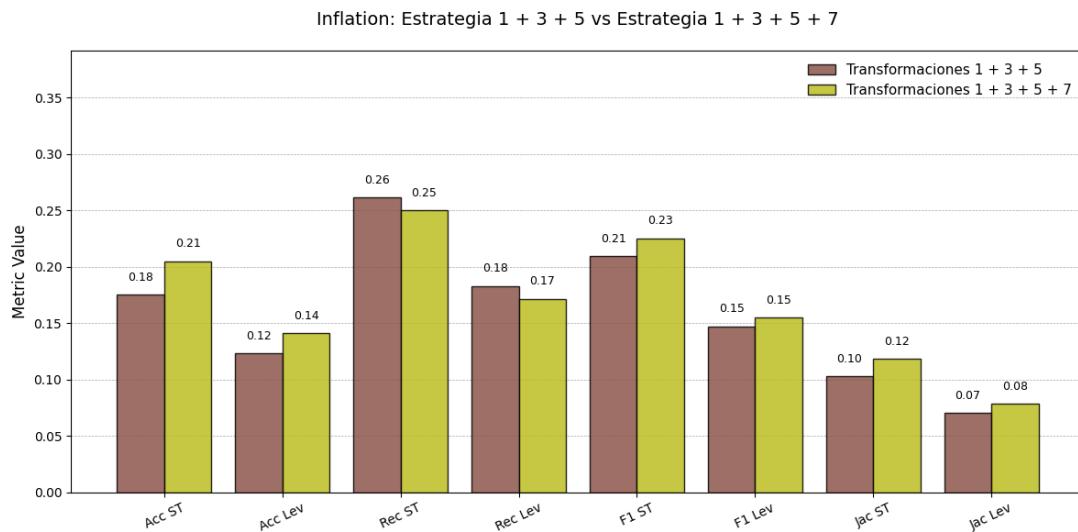


Figura 4.38: Inflation: Estrategia 1 + 3 + 5 vs Estrategia 1 + 3 + 5 + 7

Estos resultados muestran que la eliminación de tripletas redundantes mejora de forma general la precisión y la calidad del grafo, con un coste muy bajo en la cobertura en algunos casos.

#### Evolución media de las métricas

En la Figura 4.39 se aprecia que, al aplicar la Estrategia 7, todas las métricas mejoran a excepción de la cobertura. La cobertura semántica apenas se ve afectada, empeorando solo un 5 %, y la cobertura léxica disminuye únicamente un 1 %. Por tanto, la Estrategia 7 cumple su objetivo de aumentar la precisión y la coherencia estructural sin alterar de manera significativa la cobertura del grafo resultante.

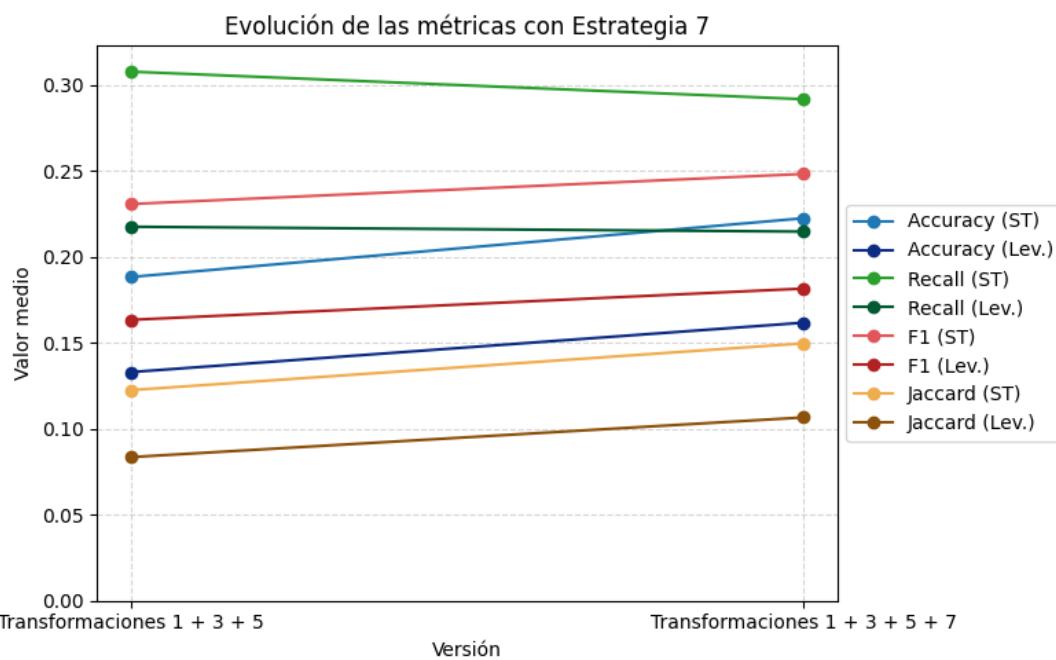


Figura 4.39: Evolución media de las métricas Estrategia 7

#### 4.2.3.3. Evaluación del Bloque III

El objetivo de este último bloque consistía en mejorar la precisión del grafo obtenido tras aplicar las transformaciones seleccionadas en los bloques anteriores.

Por un lado, la Estrategia 6 no logra mejoras constantes. En general reduce las métricas y apenas incrementa la precisión, de modo que los pocos avances obtenidos resultan insuficientes frente a las pérdidas en cobertura y similitud estructural.

Por otro lado, la estrategia 7 de eliminación de tripletas similares, sí que muestra un aumento en la precisión y la coherencia estructural sin alterar de manera significativa la cobertura.

En la Figura 4.40 se aprecia que la Estrategia 6 apenas aporta mejoras, mientras que la Estrategia 7 mejora todas las métricas salvo la cobertura, que sufre una ligera disminución respecto al grafo de inicio de este bloque. Por tanto, se selecciona la Estrategia 7, pues es la que mejor cumple el objetivo de conservar únicamente las tripletas más precisas, eliminando las redundancias.

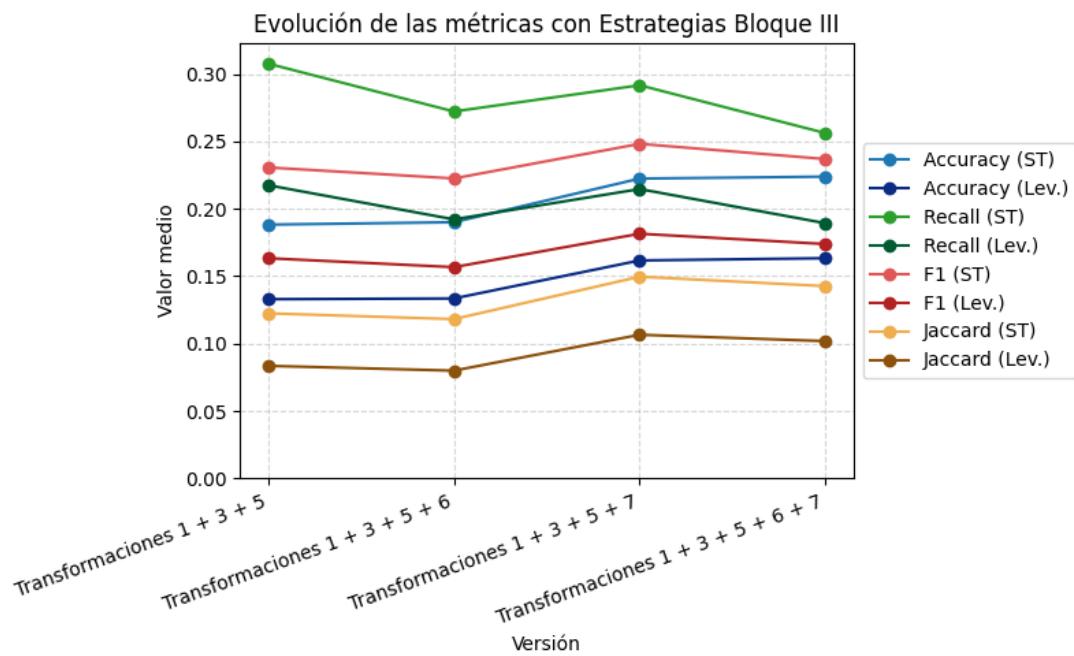


Figura 4.40: Evaluación Bloque III

### 4.3. Evaluación final: Resultado frente a otros enfoques

En esta sección se comparan las métricas del grafo Baseline inicial, del grafo resultante tras aplicar el pipeline mejorado y de los grafos generados por sistemas basados en LLMs, tomando como referencia el grafo Gold. El objetivo es ofrecer una evaluación exhaustiva del rendimiento del enfoque con recursos limitados frente al Baseline inicial y frente al enfoque de los modelos de lenguaje de última generación en términos de cobertura, precisión y similitud estructural.

#### Utilitarianism

Para *Utilitarianism*, el pipeline de mejoras eleva de forma clara la solidez del grafo respecto al punto de partida, logrando una representación más coherente y completa. En todas las métricas se observa un incremento de magnitud similar, lo que permite incluso superar al grafo generado con Neo4jBuilder en cobertura, algo que el Baseline no lograba. No obstante, pese a las mejoras evidentes en la Figura 4.41, los grafos producidos por los LLM siguen estando más alineados con el grafo Gold.

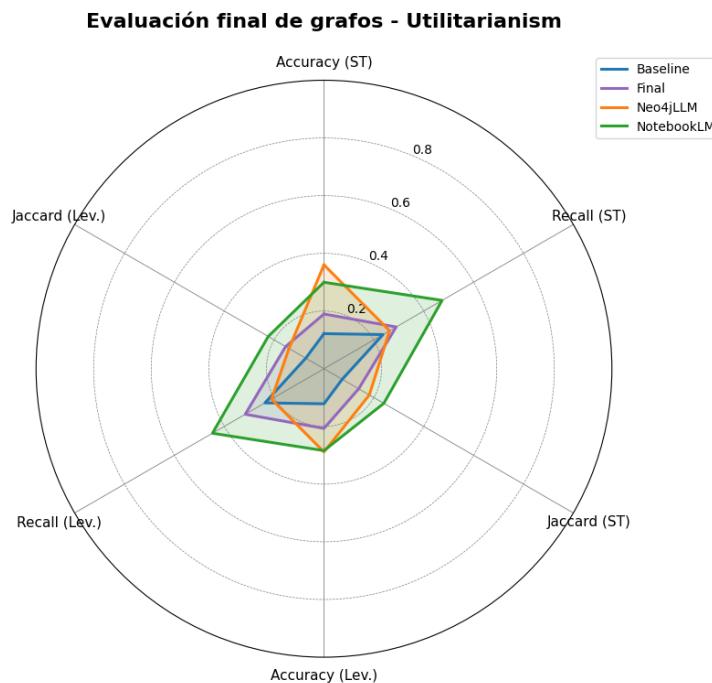


Figura 4.41: Evaluación final de grafos - Utilitarianism

#### Ellipse

En el caso de *Ellipse*, el pipeline mejorado logra un avance claro respecto al Baseline, aumentando todas las métricas. Destacan especialmente las mejoras con respecto al Baseline en el ámbito semántico, con incrementos de 58.3 % en precisión, 37.5 % en cobertura y 71.9 % en el índice de Jaccard. Frente a Neo4jBuilder, el grafo final prácticamente iguala su precisión, algo que el Baseline no conseguía. En la Figura 4.42 se aprecia cómo este enfoque con recursos limitados se acerca al rendimiento de los LLMs, ofreciendo un grafo mucho más fiel y amplio que el Baseline original.

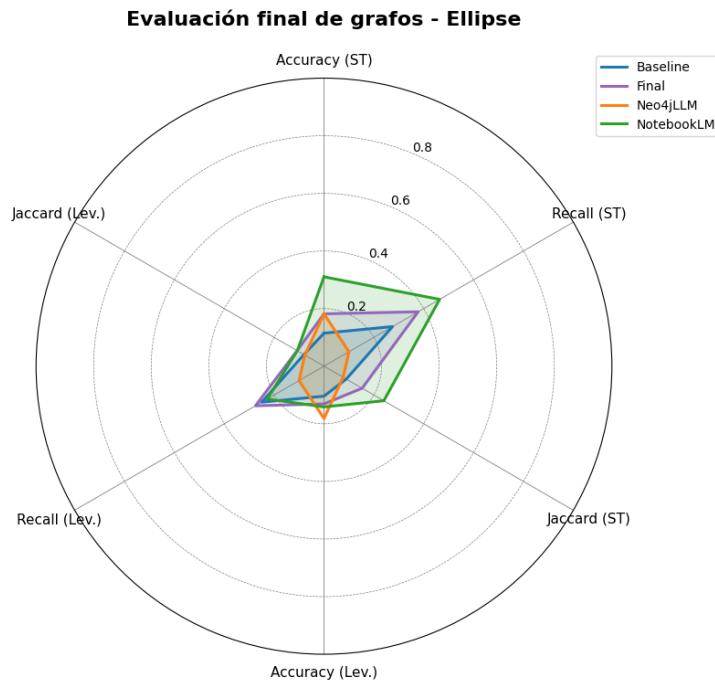


Figura 4.42: Evaluación final de grafos - Ellipse

### Homeostasis

En el caso de *Homeostasis*, las mejoras tras aplicar el pipeline no son tan consistentes como en los textos anteriores. Como muestra la Figura 4.43, las métricas semánticas experimentan un avance, mientras que las métricas léxicas se reducen ligeramente respecto al grafo Baseline. Esto indica que, aunque la aplicación de mejoras fortalece la captura de relaciones a nivel de significado, introduce cierta variabilidad en la forma de los términos, lo que hace que el grafo se aleje léxicamente del grafo Gold. Frente a los sistemas basados en LLMs, el grafo mejorado sigue quedando por detrás. Neo4jLLM ofrece la representación más precisa y estructuralmente coherente, por encima de NotebookLM, quedando el grafo Baseline y el final por debajo en todos los ámbitos.

### Inflation

En el caso del texto de la inflación, el pipeline de mejoras logra también un avance sobre el Baseline, pero sigue alejado de los grafos construidos mediante LLMs. Respecto al Baseline, la precisión del grafo final se duplica y tanto la cobertura como la estructura del grafo se refuerzan de forma clara. Aun así, nuevamente, los enfoques basados en grandes modelos de lenguaje ofrecen representaciones significativamente más precisas y completas.

### Conclusión comparación final

Como se analizó al principio, para todos los textos del dataset de referencia los enfoques basados en LLMs obtienen los mejores resultados al compararlos con el grafo Gold. En las evaluaciones de los grafos finales, se aprecia que en general los LLMs superan al grafo generado mediante las tripletas mejoradas de OpenIE en todas las métricas evaluadas tanto en sus variantes semántica como léxica.

Aunque la versión mejorada del grafo de OpenIE mejora respecto al Baseline, los grafos generados por

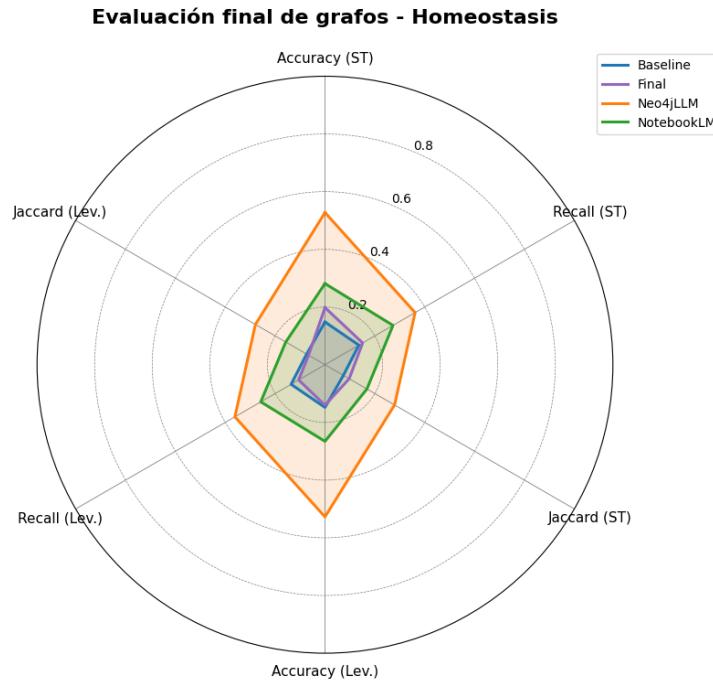


Figura 4.43: Evaluación final de grafos - Homeostasis

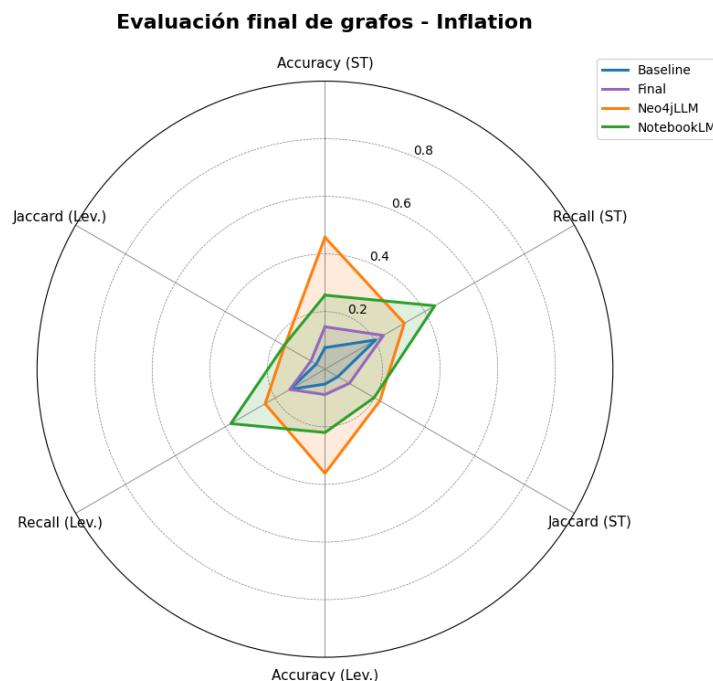


Figura 4.44: Evaluación final de grafos - Inflation

LLMs resultan conceptualmente más coherentes y se alinean con mayor fidelidad al grafo Gold. Los grafos generados por Notebook LM destacan mayormente por su cobertura y los generados por Neo4jLLM Graph builder, por su precisión. Como se vio en el análisis inicial, a pesar de las limitaciones de los enfoques basados en LLMs, estos siguen ofreciendo resultados más próximos al grafo Gold de referencia.

## 4.4. Objetivos Logrados

### 4.4.1. Objetivos Específicos

Para analizar si se cumplen los objetivos específicos planteados inicialmente, es necesario observar las mejoras en las métricas correspondientes a cada bloque de estrategias. En la Figura 4.45, se muestra la evolución de las métricas de calidad de los grafos tras las distintas transformaciones seleccionadas de cada bloque.

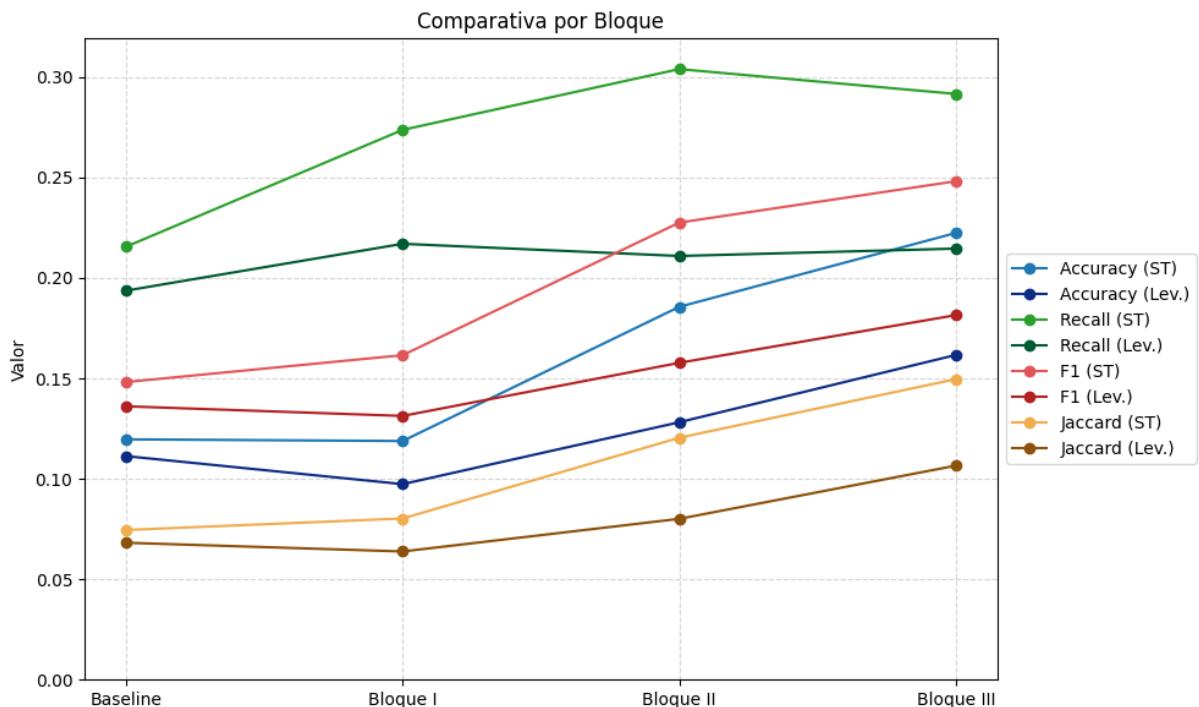


Figura 4.45: Evolución de las métricas por bloque.

#### Objetivo 1

El primer objetivo del proyecto consistía en ampliar la cobertura del grafo Baseline, extrayendo nuevas tripletas que añadieran información relevante no capturada inicialmente. Para ello, el Bloque I de estrategias se centró en este propósito, y logró incrementar tanto la cobertura semántica como la léxica, sin que ello supusiera una pérdida significativa en la precisión del grafo.

Aunque en fases posteriores se aplican transformaciones que reducen un poco la cobertura para mejorar en mayor medida la precisión, el grafo final transformado presenta una cobertura superior a la del Baseline original, lo que demuestra que el primer objetivo se cumple. Concretamente, en los grafos generados a partir del dataset de referencia, se logró una mejora media de la cobertura en un 35 % en términos semánticos y en un 11 % en términos léxicos respecto al grafo Baseline.

### Objetivo 2

El segundo objetivo del proyecto se centraba en corregir el problema de OpenIE de los límites imprecisos en las extracciones, especialmente en los predicados vagos y en la representación inconsistente de entidades. El objetivo era refinar las relaciones y unificar las entidades que hacían referencia a lo mismo, con el fin de mejorar la cohesión y la claridad del grafo.

El Bloque II fue el encargado de abordar este objetivo, y logró mejoras en todas las métricas, salvo en la cobertura léxica. Esta ligera disminución tiene sentido, ya que el grafo pasó de tener múltiples formas distintas de representar una misma entidad a contar con una única representación unificada, lo cual mejora su consistencia y facilita su interpretación.

### Objetivo 3

El tercer objetivo se centraba en reducir la redundancia o la imprecisión de tripletas presentes en el grafo. El propósito era mejorar la precisión final eliminando aquellas tripletas que, aunque distintas en forma, transmitían la misma información.

Las estrategias aplicadas en el Bloque III abordaron este problema de manera efectiva. Aunque provocaron una ligera disminución en la cobertura, consiguieron mejorar de forma notable la precisión del grafo final, cumpliendo así el tercer objetivo. La precisión semántica prácticamente se duplicó y se mejoró la precisión léxica en un 45 % con respecto al grafo Baseline.

### Objetivo 4

El último objetivo del proyecto consistía en evaluar el impacto de las distintas estrategias de mejora aplicadas al grafo Baseline. Para ello, se analizó el efecto de cada estrategia tanto a nivel individual (por texto) como a nivel global (media del dataset de referencia), interpretando las métricas obtenidas y comparando los grafos resultantes con el grafo Gold. Finalmente, se realizó una comparación del grafo generado por el pipeline completo con los enfoques basados en LLMs, con el objetivo de valorar hasta qué punto las mejoras aplicadas acercarse a los resultados de sistemas más complejos.

#### 4.4.2. Objetivo General

Este proyecto tenía como objetivo crear un pipeline que generara un grafo de conocimiento a partir de texto, utilizando únicamente OpenIE y aplicando estrategias lingüísticas y estructurales. El propósito era aproximarse al rendimiento de los sistemas de construcción de grafos de conocimiento basados en grandes modelos de lenguaje.

En el apartado “Evaluación preliminar: Baseline frente a otros enfoques” de la sección de resultados, se evidenciaron claramente las limitaciones del sistema basado únicamente en OpenIE en comparación con los enfoques que utilizan LLMs. En general, la herramienta de Google, NotebookLM, obtuvo los mejores resultados, seguida de Neo4LLM Graph Builder, y en último lugar, el sistema Baseline construido con OpenIE. Cabe destacar que OpenIE no es una herramienta diseñada específicamente para la generación de grafos de conocimiento, sino para la extracción de información a partir del texto en forma de tripletas sujeto–predicado–objeto. Por esta razón, fue necesario aplicar una serie de estrategias a dichas tripletas con el objetivo de hacerlas más concisas, precisas y coherentes, permitiendo así la construcción de un grafo de conocimiento estructurado y comparable con los generados por sistemas basados en LLMs.

Tras evidenciar las carencias del Baseline de OpenIE, se implementaron diversas estrategias orientadas

a expandir la cobertura, refinar y homogeneizar las tripletas, y mejorar su precisión. Estas estrategias transformaron significativamente el grafo generado inicialmente, convirtiéndolo en una estructura más coherente y alineada con el objetivo del proyecto. Este pipeline logró mejorar notablemente los resultados del Baseline inicial, como se detalla en el apartado “Evaluación final: resultado frente a otros enfoques” de la sección de resultados. Allí se observa que, tras aplicar el conjunto de estrategias a los textos del dataset de referencia, las métricas de evaluación de grafos comparadas con los grafos Gold muestran una mejora significativa.

En la Figura 4.46 se puede ver de forma gráfica la mejora que aportan las estrategias implementadas.

Al comparar la versión Final con el Baseline, se observa una mejora significativa en la mayoría de las métricas. La precisión semántica casi se duplica, con un incremento del 96.78 %, mientras que la precisión léxica mejora un 70.58 %. En términos de cobertura, la mejora es más moderada: la versión Final supera al Baseline en 15.38 % semánticamente, aunque se mantiene igual en el plano léxico. En cuanto a la similitud estructural con el grafo Gold, también se registran aumentos notables: un aumento del 89.27 % en el plano semántico y de un 60.38 % en el plano léxico.

Se puede concluir que el objetivo de generar un grafo de conocimiento más cercano a los obtenidos mediante LLMs se cumple, logrando mejorar en gran medida el grafo Baseline.

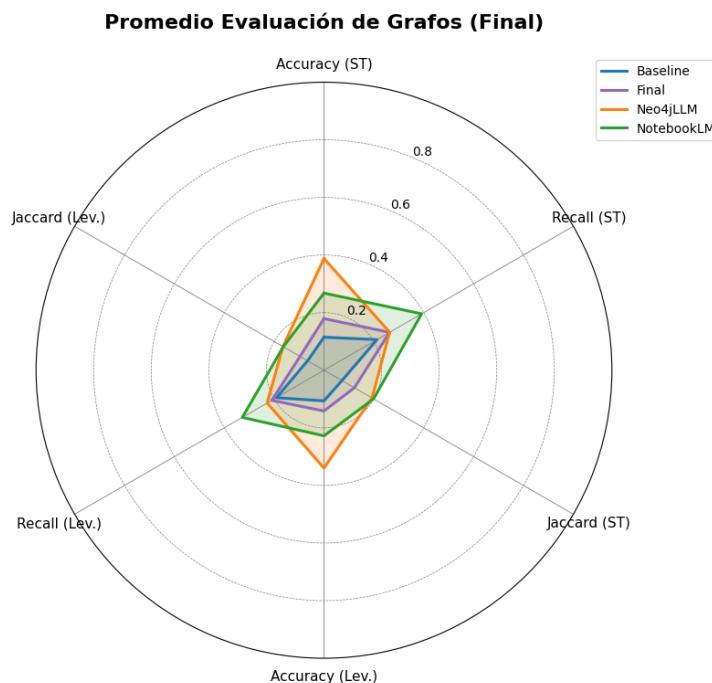


Figura 4.46: Evaluación final promedio para el dataset de referencia

Versión	Accuracy ST	Accuracy Levenshtein	Recall ST	Recall Levenshtein	Jaccard ST	Jaccard Levenshtein
Baseline	0.0747	0.0517	0.2031	0.1406	0.0518	0.0353
Final	0.1471	0.0882	0.2344	0.1406	0.0980	0.0566
Neo4jLLM	0.4577	0.3615	0.3182	0.2401	0.2200	0.1619
NotebookLM	0.2569	0.2199	0.4403	0.3778	0.1973	0.1645

## 4.5. Problemas encontrados

Durante el desarrollo de este PFG se han identificado y enfrentado diversos problemas que han influido en la calidad y rendimiento del sistema propuesto.

Uno de los principales desafíos estuvo relacionado con el uso de OpenIE de Stanford CoreNLP como herramienta base para la extracción de triplets. Tal como se detalló en la sección de estado del arte, este sistema presenta varias limitaciones. Una de las más frecuentes es la generación de triplets con límites incorrectos en los argumentos o predicados, lo que se traduce en extracciones con un alcance demasiado amplio o, en otros casos, excesivamente restringido. A pesar de que el pipeline desarrollado incorpora estrategias lingüísticas y estructurales para mitigar este problema, en algunas ocasiones las entidades resultantes siguen siendo demasiado extensas, vagas o poco informativas. Otro problema importante ha sido la redundancia en la extracción de triplets. OpenIE tiende a generar múltiples versiones de una misma relación o fragmentos de texto muy similares, lo que introduce un nivel considerable de ruido en el grafo final. Aunque las estrategias de limpieza implementadas han logrado reducir en parte estas repeticiones, eliminarlas completamente ha resultado ser una tarea muy difícil.

La elección de Stanford CoreNLP se justificó por ser una de las herramientas más recientes dentro del ecosistema de OpenIE que no depende de LLMs, lo que la hace más accesible y ligera. Sin embargo, su antigüedad representa una limitación significativa: muchos de sus componentes no se han actualizado desde 2015, y en algunos casos, los parámetros de configuración no tienen efecto a pesar de ser modificados, lo que restringe el control sobre su comportamiento. Además, su implementación en Java ha generado ciertas dificultades de integración, ya que la mayoría de las técnicas de PLN empleadas en este proyecto están desarrolladas en Python, lo que ha añadido complejidad en términos de comunicación entre herramientas.

En cuanto al comportamiento específico de OpenIE, se han detectado carencias importantes que afectan la cobertura de las triplets extraídas. Por ejemplo, las frases en modo imperativo como “*Cut a piece of string longer than the distance between the two thumbtacks*” no producen ninguna triplete. Lo mismo ocurre con frases entre paréntesis o expresadas en forma impersonal, las cuales son ignoradas por el sistema. Estas limitaciones estructurales reducen significativamente la cobertura del grafo de conocimiento generado.

Otro aspecto crítico ha sido la evaluación de la calidad de los grafos. Aunque se ha utilizado un grafo Gold de referencia para comparar los resultados, es importante tener en cuenta que, a partir de un mismo texto, pueden generarse múltiples grafos correctos desde el punto de vista conceptual. En este sentido, las métricas utilizadas presentan ciertas limitaciones, ya que un grafo puede estar conceptualmente bien construido pero no coincidir con el Gold, lo que penaliza su puntuación. Por esta razón, además de las métricas cuantitativas, ha sido necesario realizar un análisis cualitativo de los grafos generados, evaluando de forma manual aspectos como la coherencia semántica, la estructura lógica y la fidelidad respecto al contenido original del texto. Esto se debe a que la interpretación lógica y contextual de un ser humano sigue siendo clave para determinar si un grafo, más allá de obtener buenas métricas, resulta realmente comprensible y coherente. Aunque hubiese sido ideal contar con varios grafos de referencia anotados manualmente, la creación de grafos supervisados por humanos es un proceso costoso.

En resumen, aunque se han mitigado muchas de las limitaciones mediante estrategias específicamente diseñadas para mejorar la salida de OpenIE, persisten varios obstáculos que condicionan los resultados obtenidos. Por un lado, OpenIE presenta carencias estructurales que afectan a la precisión, cobertura y coherencia de las tripletas extraídas, incluyendo errores en los límites, redundancias y dificultad para manejar ciertos tipos de frases. Por otro lado, las métricas utilizadas para evaluar los grafos tienen limitaciones inherentes, ya que no siempre reflejan adecuadamente la calidad conceptual del grafo frente a múltiples posibles interpretaciones válidas. Además, la decisión de no utilizar grandes recursos supone una desventaja notable en un contexto actual en el que los LLMs dominan el panorama del PLN y ofrecen un rendimiento muy superior en tareas complejas como la construcción de grafos de conocimiento.

# Capítulo 5

# Conclusiones y trabajos futuros

## 5.1. Conclusiones

Este proyecto ha abordado la generación de grafos de conocimiento a partir de texto desde una perspectiva basada en no usar grandes recursos.

Para ello, se partió de un grafo base generado mediante OpenIE, una herramienta de código abierto para la extracción de triplets sujeto-predicado-objeto a partir de texto. A partir de la comparación inicial del grafo Baseline con un grafo Gold (creado manualmente) y con dos aproximaciones basadas en LLMs (Neo4j LLM Graph Builder y Notebook LM), se identificaron importantes limitaciones del enfoque con OpenIE en términos de precisión y cobertura.

Con el objetivo de mejorar el grafo Baseline, se diseñó y aplicó un conjunto de estrategias distribuidas en tres bloques: (1) expansión de cobertura, (2) homogeneización y refinamiento, y (3) mejora de precisión. Estas estrategias incluyeron técnicas como el NER, la re-extracción sobre fragmentos verbales, la limpieza de sujetos y objetos con técnicas de mapeo, o el filtrado de triplets poco informativas o redundantes.

Los resultados muestran que la aplicación de estas estrategias permitió mejorar significativamente las métricas evaluadas (precisión, cobertura y similitud estructural con el grafo Gold), reduciendo la brecha entre el enfoque basado en OpenIE y los sistemas que utilizan LLMs, sin aplicar transformaciones muy costosas.

Se puede afirmar que el objetivo general del proyecto se ha cumplido: se ha logrado diseñar y validar un pipeline ligero para la generación de grafos de conocimiento a partir de texto sin recurrir a grandes modelos de lenguaje. Este pipeline, basado en OpenIE y enriquecido mediante estrategias de bajo coste computacional, ha demostrado mejoras notables respecto al sistema inicial. Asimismo, se han alcanzado los objetivos específicos, con resultados positivos en cada uno de los bloques de estrategias propuestas.

No obstante, el rendimiento del sistema mejorado no llega a igualar al de los enfoques basados en LLMs, ya que el KGC es una tarea que requiere una gran comprensión semántica y manejo contextual avanzado. Aun así, los resultados obtenidos representan un avance significativo respecto a la base original, y demuestran que es posible reducir la brecha de calidad mediante técnicas accesibles y de bajo coste computacional.

## 5.2. Impacto social y medioambiental

Al desarrollar soluciones tecnológicas, no solo deben considerarse los objetivos técnicos específicos, sino también las posibles repercusiones sociales y medioambientales que pueden derivarse de su aplicación. Es fundamental reflexionar sobre cómo estas herramientas afectan tanto a la sociedad como al entorno natural.

En primer lugar, desde el punto de vista social, este proyecto centrado en la generación de grafos de conocimiento a partir de texto sin recurrir a modelos de lenguaje de gran escala, trata de proponer una alternativa más accesible para contextos educativos, investigadores independientes o instituciones

con recursos computacionales limitados. Además, cabe destacar que los grafos de conocimiento tienen un enorme potencial en dominios específicos, como la medicina o la educación, donde transformar información no estructurada en representaciones estructuradas puede motivar aplicaciones con un impacto significativo. Por tanto, el enfoque presentado no solo ofrece beneficios técnicos, sino también un aporte real a la mejora de procesos en ámbitos sensibles para la sociedad.

En cuanto al impacto medioambiental, este trabajo responde a una preocupación creciente: el elevado coste energético y la huella de carbono derivados del uso masivo de modelos de lenguaje de gran escala. Al proponer un pipeline más ligero y eficiente, se contribuye a reducir el consumo de recursos y se apuesta por un desarrollo tecnológico más sostenible.

Este PFG se alinea con varios Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030 [43]. Principalmente, con el ODS 9: Industria, Innovación e Infraestructura, al fomentar la creación de soluciones innovadoras y sostenibles accesibles para todo tipo de usuarios. También se relaciona con el ODS 12: Producción y Consumo Responsables, al evitar el uso de sistemas computacionalmente intensivos y promover una mayor eficiencia energética.



Figura 5.1: ODS relacionados con este PFG

### 5.3. Líneas futuras

A partir del análisis realizado en este proyecto, se identifican diversas líneas futuras de trabajo que pueden contribuir a avanzar en la generación eficiente de grafos de conocimiento a partir de texto.

Es innegable que los grandes modelos de lenguaje no solo representan el presente, sino también el futuro de muchas tareas relacionadas con el KGC. No obstante, el pipeline explorado en este trabajo demuestra que las técnicas tradicionales aún tienen recorrido, especialmente en contextos con restricciones de recursos. Aunque no siempre representan el camino más directo, pueden aportar mejoras interesantes si se combinan de forma inteligente.

Una posible línea futura sería la combinación de técnicas ligeras como OpenIE con el potencial de los LLMs, por ejemplo mediante el fine-tuning de modelos más compactos para tareas específicas de extracción de conocimiento, procurando mitigar uno de sus principales problemas: las alucinaciones. También sería interesante explorar modelos intermedios o sistemas híbridos basados en reglas y aprendizaje automático, que podrían ofrecer un equilibrio entre precisión y eficiencia.

En cuanto al pipeline desarrollado, se abren múltiples oportunidades de mejora. Por un lado, su adaptación a otros idiomas o dominios especializados permitiría evaluar su robustez y generalización. Por otro, podrían incorporarse técnicas más avanzadas de PLN, como el enriquecimiento de relaciones mediante recursos externos como Wikidata [44].

Además, para el problema de las relaciones redundantes o poco informativas, una línea futura relevante

sería aplicar técnicas de agrupamiento o clusterización semántica de relaciones, con el fin de mejorar la calidad estructural y conceptual del grafo. Otra posible mejora sería el desarrollo de una interfaz de usuario sencilla que permitiera cargar un texto y visualizar automáticamente el grafo generado, ocultando al usuario la complejidad técnica del pipeline y facilitando su adopción en contextos reales.

Por último, también resulta de gran interés la generación de datasets específicos para tareas de construcción de grafos de conocimiento a partir de texto, especialmente aquellos que incluyan múltiples grafos de referencia (grafos Gold) por texto. Este tipo de recursos permitiría tanto una evaluación más rica como la posibilidad de entrenar modelos especializados más allá de los LLMs multitarea genéricos.

# Bibliografía

- [1] X. Zou, “A survey on application of knowledge graph,” in *Journal of Physics: Conference Series*, vol. 1487, no. 1. IOP Publishing, 2020, p. 012016. [Online]. Available: <https://iopscience.iop.org/article/10.1088/1742-6596/1487/1/012016/meta>
- [2] D. Xu, W. Chen, W. Peng, C. Zhang, T. Xu, X. Zhao, X. Wu, Y. Zheng, Y. Wang, and E. Chen, “Large language models for generative information extraction: A survey,” *Frontiers of Computer Science*, vol. 18, no. 6, p. 186357, 2024. [Online]. Available: <https://link.springer.com/article/10.1007/s11704-024-40555-y>
- [3] G. Angeli, M. J. J. Premkumar, and C. D. Manning, “Leveraging linguistic structure for open domain information extraction,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 344–354. [Online]. Available: <https://aclanthology.org/P15-1034.pdf>
- [4] Neo4j Labs, “Neo4j LLM Knowledge Graph Builder - Extract Nodes and Relationships from Unstructured Text,” 2024. [Online]. Available: <https://neo4j.com/labs/genai-ecosystem/llm-graph-builder/>
- [5] Q. Wang, Z. Mao, B. Wang, and L. Guo, “Knowledge graph embedding: A survey of approaches and applications,” *IEEE transactions on knowledge and data engineering*, vol. 29, no. 12, pp. 2724–2743, 2017. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8047276>
- [6] R. Handley, “Google Knowledge Graph: What It Is & Why It Matters,” 2024. [Online]. Available: <https://www.semrush.com/blog/knowledge-graph/>
- [7] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, “A survey on knowledge graphs: Representation, acquisition, and applications,” *IEEE transactions on neural networks and learning systems*, vol. 33, no. 2, pp. 494–514, 2021. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9416312>
- [8] P. Liu, W. Gao, W. Dong, L. Ai, Z. Gong, S. Huang, Z. Li, E. Hoque, J. Hirschberg, and Y. Zhang, “A survey on open information extraction from rule-based model to large language model,” *arXiv preprint arXiv:2208.08690*, 2022. [Online]. Available: <https://arxiv.org/abs/2208.08690>
- [9] Neo4j Labs, “Introduction to the Neo4j LLM Knowledge Graph Builder,” 2024. [Online]. Available: <https://neo4j.com/blog/developer/llm-knowledge-graph-builder/>
- [10] Google Labs, “Introducing Notebook LM,” 2023. [Online]. Available: <https://blog.google/technology/ai/notebooklm-google-ai/>
- [11] Renée Cho, “AI’s Growing Carbon Footprint,” 2023. [Online]. Available: <https://news.climate.columbia.edu/2023/06/09/ais-growing-carbon-footprint/>
- [12] H. Ye, N. Zhang, H. Chen, and H. Chen, “Generative knowledge graph construction: A review,” *arXiv preprint arXiv:2210.12714*, 2022. [Online]. Available: <https://arxiv.org/pdf/2210.12714>
- [13] I. Melnyk, P. Dognin, and P. Das, “Knowledge graph generation from text,” *arXiv preprint arXiv:2211.10511*, 2022. [Online]. Available: <https://arxiv.org/abs/2211.10511>
- [14] J. You, R. Ying, X. Ren, W. Hamilton, and J. Leskovec, “Graphrnn: Generating realistic graphs with deep auto-regressive models,” in *International conference on machine learning*. PMLR, 2018, pp. 5708–5717. [Online]. Available: <https://proceedings.mlr.press/v80/you18a.html>

- [15] B. Xue and L. Zou, "Knowledge graph quality management: A comprehensive survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 5, pp. 4969–4988, 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9709663>
- [16] A. Yates, M. Banko, M. Broadhead, M. J. Cafarella, O. Etzioni, and S. Soderland, "Textrunner: open information extraction on the web," in *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2007, pp. 25–26. [Online]. Available: <https://aclanthology.org/N07-4013.pdf>
- [17] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and M. Mausam, "Open information extraction: The second generation." in *IJCAI*, vol. 11, 2011, pp. 3–10. [Online]. Available: <https://aiweb.cs.washington.edu/research/projects/aiweb/media/papers/etzioni-ijcai2011.pdf>
- [18] M. Schmitz, S. Soderland, R. Bart, O. Etzioni *et al.*, "Open language learning for information extraction," in *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, 2012, pp. 523–534. [Online]. Available: <https://aclanthology.org/D12-1048.pdf>
- [19] L. Del Corro and R. Gemulla, "Clausie: clause-based open information extraction," in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 355–366. [Online]. Available: <https://dl.acm.org/doi/pdf/10.1145/2488388.2488420>
- [20] P. Liu, W. Gao, W. Dong, S. Huang, and Y. Zhang, "Open information extraction from 2007 to 2022—a survey," *arXiv preprint arXiv:2208.08690*, 2022. [Online]. Available: [https://www.researchgate.net/publication/362788554\\_Open\\_Information\\_Extraction\\_from\\_2007\\_to\\_2022\\_-\\_A\\_Survey](https://www.researchgate.net/publication/362788554_Open_Information_Extraction_from_2007_to_2022_-_A_Survey)
- [21] Y. Zhu, X. Wang, J. Chen, S. Qiao, Y. Ou, Y. Yao, S. Deng, H. Chen, and N. Zhang, "Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities," *World Wide Web*, vol. 27, no. 5, p. 58, 2024. [Online]. Available: <https://link.springer.com/article/10.1007/s11280-024-01297-w>
- [22] B. Chen and A. L. Bertozzi, "Autokg: Efficient automated knowledge graph generation for language models," in *2023 IEEE International Conference on Big Data (BigData)*. IEEE, 2023, pp. 3117–3126. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10386454>
- [23] R. Glauber and D. B. Claro, "A systematic mapping study on open information extraction," *Expert Systems with Applications*, vol. 112, pp. 372–387, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0957417418303932>
- [24] R. Schneider, T. Oberhauser, T. Klatt, F. A. Gers, and A. Löser, "Analysing errors of open information extraction systems," *arXiv preprint arXiv:1707.07499*, 2017. [Online]. Available: <https://arxiv.org/abs/1707.07499>
- [25] M. Trajanoska, R. Stojanov, and D. Trajanov, "Enhancing knowledge graph construction using large language models," *arXiv preprint arXiv:2305.04676*, 2023. [Online]. Available: <https://arxiv.org/abs/2305.04676>
- [26] N. Mihindukulasooriya, S. Tiwari, C. F. Enguix, and K. Lata, "Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text," in *International semantic web conference*. Springer, 2023, pp. 247–265. [Online]. Available: [https://link.springer.com/chapter/10.1007/978-3-031-47243-5\\_14](https://link.springer.com/chapter/10.1007/978-3-031-47243-5_14)
- [27] J. Sun, S. Qian, Z. Han, W. Li, Z. Qian, D. Yang, J. Cao, and G. Xue, "Lkd-kgc: Domain-specific kg construction via llm-driven knowledge dependency parsing," *arXiv preprint arXiv:2505.24163*, 2025. [Online]. Available: <https://arxiv.org/abs/2505.24163>

- [28] P. d. M. Ligabue, A. A. F. Brandão, S. M. Peres, F. G. Cozman, and P. Pirozelli, “Applying a context-based method to build a knowledge graph for the blue amazon,” *Data Intelligence*, vol. 6, no. 1, pp. 64–103, 2024. [Online]. Available: <https://direct.mit.edu/dint/article/6/1/64/119998>
- [29] J. L. Martinez-Rodriguez, I. Lopez-Arevalo, and A. B. Rios-Alvarado, “Openie-based approach for knowledge graph construction from text,” *Expert Systems with Applications*, vol. 113, pp. 339–355, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417418304329>
- [30] J. L. Martinez-Rodriguez, A. Hogan, and I. Lopez-Arevalo, “Information extraction meets the semantic web: a survey,” *Semantic Web*, vol. 11, no. 2, pp. 255–335, 2020. [Online]. Available: <https://journals.sagepub.com/doi/full/10.3233/SW-180333>
- [31] J. Driver, “The History of Utilitarianism,” in *The Stanford Encyclopedia of Philosophy*, winter 2022 ed., E. N. Zalta and U. Nodelman, Eds. Metaphysics Research Lab, Stanford University, 2022. [Online]. Available: <https://plato.stanford.edu/archives/win2022/entries/utilitarianism-history/>
- [32] OpenStax, *The Ellipse*, 2nd ed. OpenStax, Rice University, 2022, ch. 12.1. [Online]. Available: <https://openstax.org/books/algebra-and-trigonometry-2e/pages/12-1-the-ellipse>
- [33] Wikipedia contributors, “Homeostasis,” 2025. [Online]. Available: <https://en.wikipedia.org/wiki/Homeostasis>
- [34] tutor2u, “What is demand-pull inflation?” 2023. [Online]. Available: <https://www.tutor2u.net/economics/reference/what-is-demand-pull-inflation>
- [35] Flair NLP Project, “Flair Embeddings Tutorial,” [Online; accessed June 2025], 2025. [Online]. Available: <https://flairnlp.github.io/docs/tutorial-embeddings/flair-embeddings>
- [36] Hugging Face, “all-MiniLM-L6-v2,” [Online; accessed June 2025], 2025. [Online]. Available: <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>
- [37] Natural Language Toolkit Project, “NLTK: Natural Language Toolkit.” [Online]. Available: <https://www.nltk.org/>
- [38] ——, “nltk.metrics.distance — Distance Metrics.” [Online]. Available: <https://www.nltk.org/api/nltk.metrics.distance.html>
- [39] Wikipedia contributors, “Jaccard index.” [Online]. Available: [https://en.wikipedia.org/wiki/Jaccard\\_index](https://en.wikipedia.org/wiki/Jaccard_index)
- [40] Stanford NLP Group, “edu.stanford.nlp.dcoref.CorefChain – Stanford JavaNLP 3.5.0 Javadoc,” n.d. [Online]. Available: <https://nlp.stanford.edu/nlp/javadoc/javanlp-3.5.0/edu/stanford/nlp/dcoref/CorefChain.html>
- [41] ——, “Stanford CoreNLP OpenIE: Description,” n.d. [Online]. Available: <https://stanfordnlp.github.io/CoreNLP/openie.html#description>
- [42] ——, “Stanford CoreNLP OpenIE,” n.d. [Online]. Available: <https://stanfordnlp.github.io/CoreNLP/openie.html>
- [43] Pacto Mundial, “¿Qué puedes hacer tú? — ODS,” [Online; accessed June 2025], n.d. [Online]. Available: <https://www.pactomundial.org/que-puedes-hacer-tu/ods/>
- [44] Wikidata contributors, “Wikidata: Main Page,” n.d. [Online]. Available: [https://www.wikidata.org/wiki/Wikidata:Main\\_Page](https://www.wikidata.org/wiki/Wikidata:Main_Page)
- [45] V. M. Dominguez Rivas, *Plantilla TFG ETSISI UPM*. ETSISI, 2020.

- [46] E. G. de Heredia Solana, “Alternativa ligera a los llms en la generación de grafos de conocimiento: Mejoras a partir de openie,” 2025. [Online]. Available: <https://github.com/eliaghs/tfg-efficient-kg-generation>