UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

FACULTAD DE INGENIERÍA Y ARQUITECTURA

MAESTRÍA EN CIENCIAS Y TECNOLOGÍAS DE LA COMPUTACIÓN

Método de enriquecimiento de grafos de conocimiento basado en la inferencia de entidades semánticas equivalentes desde fuentes de datos abiertos (Enriching a knowledge graph from open knowledge datasources)

Tesis previo a la obtención del título de:

MAGÍSTER EN CIENCIAS Y TECNOLOGÍAS DE LA COMPUTACIÓN

Autor: Cárdenas Cabrera, Ana Cristina

Director: Piedra Pullaguari, Nelson Oswaldo

LOJA

2022

2022

Aprobación del director del trabajo de titulación

Loja, 16 de marzo del 2022

PhD.
Rommel Vicente Torres Tandazo
Coordinador de la Maestría en Ciencias y Tecnologías de la Computación
Ciudad.-

De mi consideración:

Me permito comunicar que, en calidad de director de la presente tesis denominado: (nombre del trabajo) realizado por Nombres y Apellidos completos del autor o autores (as) ha sido orientado y revisado durante su ejecución, así mismo ha sido verificado a través de la herramienta de similitud académica institucional, y cuenta con un porcentaje de coincidencia aceptable. En virtud de ello, y por considerar que el mismo cumple con todos los parámetros establecidos por la Universidad, doy mi aprobación a fin de continuar con el proceso académico correspondiente.

Particular que comunico para los fines pertinentes.

Atentamente,

Firma del Director del Trabajo de Titulación Nelson
Oswaldo Piedra Pullaguari, PhD.
C.I: 1102809462
Correo electrónico: nopiedra@utpl.edu.ec

Declaración de AUTORÍA y cesión de derechos

Yo, Ana Cristina Cárdenas Cabrera, declaro y acepto en forma expresa lo siguiente:

Ser autor (a) de la tesis denominado: Método de enriquecimiento de grafos de conocimiento basado en la inferencia de entidades semánticas equivalentes desde fuentes de datos abiertos (Enriching a knowledge graph from open knowledge datasources), de la maestría de CIENCIAS Y TECNOLOGÍAS DE LA COMPUTACIÓN específicamente de los contenidos comprendidos en: (se debe colocar los nombres de los capítulos elaborados en la tesis), siendo (nombres y apellidos completos), director (a) del presente trabajo; también declaro que la presente investigación no vulnera derechos de terceros ni utiliza fraudulentamente obras preexistentes. Además, ratifico que las ideas, criterios, opiniones, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad. Eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones judiciales o administrativas, en relación a la propiedad intelectual de este trabajo.

Que la presente obra, producto de mis actividades académicas y de investigación, forma parte del patrimonio de la Universidad Técnica Particular de Loja, de conformidad con el artículo 20, literal j), de la Ley Orgánica de Educación Superior; y, artículo 91 del Estatuto Orgánico de la UTPL, que establece: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado que se realicen a través, o con el apoyo financiero, académico o institucional (operativo) de la Universidad”, en tal virtud, cedo a favor de la Universidad Técnica Particular de Loja la titularidad de los derechos patrimoniales que me corresponden en calidad de autor/a, de forma incondicional, completa, exclusiva y por todo el tiempo de su vigencia.

La Universidad Técnica Particular de Loja queda facultada para ingresar el presente trabajo al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública, en cumplimiento del artículo 144 de la Ley Orgánica de Educación Superior.

Firma: .....

Autor: Ana Cristina Cárdenas Cabrera

C.I.: 1105213605

Dedicatoria

Dedico de manera especial a mi madre y padre, quien son el pilar fundamental en mi vida personal y profesional, sentando las bases de responsabilidad y deseos de superación con su ejemplo.

A mis hermanos, que con su ejemplo son el motivo principal para seguir mejorando cada día.

Agradecimiento

Agradezco a toda mi familia por el apoyo brindado en cada aspecto de mi vida, han sido un pilar fundamental en mi vida profesional, gracias por ser mi guía.

A todos los docentes que intervinieron en el transcurso de la maestría, y especialmente a Nelson Piedra quien me impulso dando ánimos en cada reunión mantenida.

¡Muchas gracias!

Índice de Contenido

Contenido

Índice de Tablas

Índice de Figuras

Resumen

El presente estudio aborda el enriquecimiento de grafos de conocimiento, específicamente en RDF. Los Grafos de Conocimiento (Knowledge Graphs) se han
convertido en un componente cada vez más crucial en los sistemas de inteligencia artificial, potenciando a asistentes digitales e inspirando varios proyectos de transformación digital a gran escala.

Como primera fase, esta investigación realiza la revisión sistemática del estado del arte en bases de datos científicas, enfocando los resultados en técnicas o métodos actuales de enriquecimiento de grafos.

En la segunda fase, el trabajo se enfoca en investigar conceptos fundamentales, relacionados a la Web Semántica y los grafos de conocimiento.

En la tercera fase, la investigación detalla el problema, objetivos y propuesta. Se describen un modelo para enriquecer grafos utilizando procesamiento de lenguaje natural tomando en cuenta el uso de fuentes de datos abiertos.

Finalmente, el estudio implementa el modelo propuesto que consta de cinco fases, las cuales se desarrollan sobre el lenguaje de programación Python haciendo uso de bibliotecas y herramientas para su construcción

Palabras clave: Grafos de conocimiento, Web Semántica, RDF, Lenguaje de procesamiento natural, enriquecimiento.

Abstract

The present study addresses the enrichment of knowledge graphs, specifically in RDF. Knowledge Graphs have become an increasingly crucial component in artificial intelligence systems, empowering digital assistants and inspiring several large-scale digital transformation projects.

As a first phase, this research performs the systematic review of the state of the art in scientific databases, focusing the results on current graph enrichment techniques or methods.

In the second phase, the work focuses on investigating fundamental concepts related to the Semantic Web and knowledge graphs.

In the third phase, the research details the problem, objectives and proposal. It describes a model to enrich graphs using natural language processing taking into account the use of open data sources.

Finally, the study implements the proposed model consisting of five phases, which are developed on the Python programming language using libraries and tools for its construction.

Keywords: Knowledge graphs, Semantic Web, RDF, Natural Processing Language, enrichment.

Introducción

Los grafos de conocimiento son representaciones estructuradas de información, la estructura de un grafo de conocimiento son los nodos y aristas, cada una de ellas almacena información de entidades y relaciones (Manzano, 2015). Su principal capacidad es modelar datos estructurados, facilitando la legibilidad de las maquinas. Actualmente los grafos de conocimientos (KG) se utilizan en diversos dominios como la inteligencia artificial y sistemas de recomendación. Según (Galkin, 2020) afirma que existen dos formas respaldadas por los principales proveedores del mercado para representar el conocimiento en grafos: Basados en RDF y Grafos de propiedades etiquetadas.

Las principales características de los grafos de conocimiento son: Mayor rendimiento y capacidad en comparación de realizar consultas en bases de datos SQL, alta escalabilidad a grandes volúmenes de datos, esquema flexible para incorporar el conocimiento y capacidad de inferencia de conocimiento (Saorín, 2019)

En respuesta al auge del uso de grafos de conocimiento surgen fuentes de datos abiertos que son de gran aporte para mantener la información conectada y actualizada. Sin embargo, “enriquecer” o “generar conocimiento” en un KG es una tarea compleja, el hecho de que un grafo de conocimiento este enriquecido semánticamente, implica que hay un significado asociado a las entidades en el grafo, teniendo como base una ontología. El problema actual es la carencia de vínculos o referencias a fuentes de datos abiertos, produciendo silos de información que no aportan ningún valor a los datos.

En este contexto, la investigación pretende enriquecer grafos de conocimiento, conectando la información con fuentes de datos abiertos, analizando lo enfoques existentes de enriquecimiento de grafos, con el fin de evaluar y optimizar estudios ya realizados, el resultado final es aportar un marco de trabajo para enriquecer cualquier grafo de conocimiento RDF.

En base a expuesto en secciones anteriores, se plantea construir el modelo de enriquecimiento de grafos desde Wikipedia como fuente de datos abiertos. En la propuesta se plantea que los grafos de conocimiento se enriquezcan agregando nuevas relaciones.

Estructura del documento

El presente trabajo de fin de master se estructura en cinco capítulos los cuales hacen referencia a agrupaciones de contenido relevante de la investigación:

En el **Capítulo 1,** denominado **Marco teórico** se definen los conceptos base relacionados con los grafos de conocimiento, técnicas de enriquecimiento, datos abiertos y herramientas. El objetivo principal del capítulo es proporcionar bases sólidas para la implementación del modelo planteado.

En el **Capítulo 2,** denominado **Estado del arte** se realiza la revisión sistemática de la investigación, en donde se describe la metodología utilizada y los resultados de seleccionar y evaluar los trabajos existentes del enriquecimiento de grafos, asimismo, se evalúa el estado actual y dominios ya investigados.

En el **Capítulo 3,** denominado **Problema** se describe el contexto de la investigación y los objetivos, del mismo modo se plantea el problema en el que esta investigación centra su análisis. Además, se enumeran los entregables del trabajo en base a los objetivos y

En el **Capítulo 4,** denominado **Propuesta** Se detalla la propuesta, las etapas y herramientas que se utilizan para el desarrollo de una solución óptima al problema descrito.

En el **Capítulo 5**, denominado Desarrollo de la propuesta se explica la construcción de cada una de las fases del método propuesto para alcanzar los objetivos, además, se ejemplifican casos de uso para la validación del marco de trabajo, también, se realiza el análisis de los resultados obtenidos en Capítulo 4 generando una discusión del modelo.

Capítulo uno

Estado del arte

En el presente capítulo se describe el análisis de los trabajos con mayor similitud a la presente investigación, a continuación, se detalla cada una de las secciones del capítulo:

1.1 Revisión sistemática: Se realiza la introducción a la investigación, describiendo: los trabajos relacionados, los retos y desafíos actuales.

1.2 Metodología: Se describe la metodología utilizada para recopilar información de los trabajos relacionados a la investigación.

1.3 Conclusiones: Se realiza una discusión de los trabajos relacionados encontrados.

1.1 Revisión sistemática

El rápido crecimiento de la Web 2.0 ha provocado que gran parte de información carezca de: estructura, semántica e integración, sin embargo; dentro del universo de la Web existen fuentes de información abierta que permite ser utilizada y distribuida atribuyendo a uno de los principios de linked data.

Los datos abiertos, han dado lugar a los grafos de conocimiento, que se han convertido en un componente crucial en los sistemas de inteligencia artificial e inspirando varios proyectos de transformación digital a gran escala, En los últimos años existe un incrementando en la investigación de técnicas avanzadas de extracción y enriquecimiento de información de los KG.

Los grafos de conocimiento desempeñan un papel muy importante; pese a el rápido crecimiento de la información hace que gran parte de bases de conocimientos se vuelvan incompletas.

Uno de los desafíos más grandes es estructurar con precisión el conocimiento, esto implica un enriquecimiento racional en los grafos, asegurando la calidad sintáctica y semántica de la información.

Para dar solución al problema de enriquecimiento, es necesario hacer uso de procesos automáticos y semiautomáticos que ayuden a gestionar de manera adecuada todo el proceso de enriquecimiento de un KG, encontrando hechos faltantes mediante los triples existentes desde fuentes de datos abiertos.

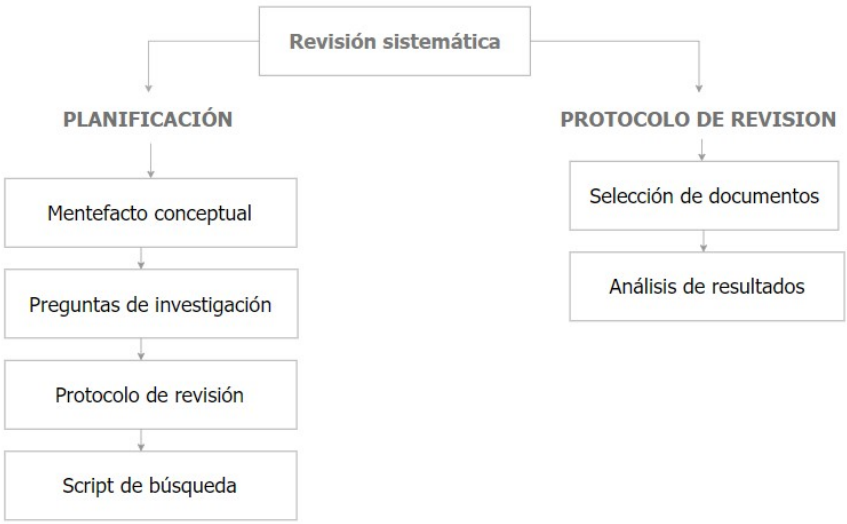
1.2 Metodología

Para recopilar y sintetizar el estado actual de la investigación acerca del enriquecimiento de los grafos, se realiza una revisión sistemática siguiendo los lineamientos planteados por (Torres-Carrion, Gonzalez-Gonzalez, Aciar, & Rodriguez-Morales, 2018) en su artículo denominado “Metodología para la revisión sistemática de la literatura aplicado a la ingeniería y educación”.

En la Figura 1 se conceptualiza la metodología aplicada, que se divide en dos fases principales: planificación y protocolo de revisión, cada una de ellas contiene sub fases para su desarrollo, a continuación, se describen cada uno de los principales ítems de la metodología:

Figura 1

Conceptualización de metodología



En los siguientes apartados se sigue lineamientos de la metodología seleccionada, cada una de las fases proporcionadas tiene como objetivo tener un resultado con el fin de realizar la selección de investigaciones similares.

1.2.1 Preguntas de investigación

Establecer las preguntas de investigación es la fase principal debido a que aumenta la probabilidad de encontrar una buena solución al objetivo planteado, en el presente trabajo se definen cuatro preguntas de investigación:

- **RQ1:** ¿Cuáles son los problemas actuales que desafían a los grafos de conocimiento?
- **RQ2:** ¿Qué criterios o técnicas existen para enriquecer un grafo de conocimiento?
- **RQ3:** ¿Cuál es el grafo de conocimiento abierto más relevante en la web?
- **RQ4:** ¿Cuáles son los dominios que han sido trabajados para enriquecer un KG?

1.2.2 Protocolo de revisión

En esta fase se construye el script de búsqueda para identificar los trabajos relacionados existentes en el área. Asimismo, permite profundizar el problema actual y contextualizar los resultados obtenidos. Para el trabajo se realiza la búsqueda en la base de conocimiento Scopus.

Para la construcción del script se detallan los criterios generales, específicos y de exclusión:

- **Criterios generales:** Las investigaciones se deben enfocar en el enriquecimiento de grafos de conocimiento, publicados desde el año 2016.
- **Criterios específicos:** Las investigaciones deben de hacer uso de técnicas de enriquecimiento de grafos de conocimiento e introducción a la predicción de enlaces.

- **Criterios de exclusión:** No se utilizarán revistas con idioma diferente al inglés o español y el tipo de publicación es tipo journal.

1.2.3 Estructura semántica

En la fase de construcción del script de búsqueda se utiliza los criterios mencionados en la Sección 1.2.2 denominado Protocolo de revisión, en la Tabla 1 se realiza la búsqueda de los sinónimos al tema a investigar usando palabras del tesauro.

Tabla 1

Palabras desde tesauro

Knowledge Graph	Semantic enrichment method	Open data
Knowledge graph **Semantic link***	**Enrichment and (method, Process, Technique, Mechanism)**	Open, accessible and (data , info*) (sources)

Se toma como base los sinónimos descritos en la Tabla 1 y se procede a la construcción de la estructura semántica, en la Tabla 2 se especifican tres niveles que se acoplarán a diferentes bases de datos científicas .

Tabla 2

Estructura semántica

Nivel	Tema	Estructura de búsqueda
N1	**Grafos de conocimiento**	(knowledge OR semantic) W/2(graph OR link)
N2	**Método de enriquecimiento**	AND (enrichment AND (method OR process OR technique OR Mechanism))
N3	**Datos abiertos**	AND (Open OR accessible) W/2 (data OR info) and (sources)

de búsqueda, el objetivo principal de esta sección es construir un script solido que abarque los temas relevantes de la investigación. En la Tabla 3 se presentan el script final con los criterios generales mencionados en la sección 1.2.3.

Tabla 3

Desarrollo de Script de búsqueda

Bases de datos	**Cadena de búsqueda**
- Scopus **-Web Of Science**	TITLE-ABS-KEY ((semantic OR enrichment) AND (method OR process OR technique OR mechanism) AND (knowledge) W/2 (graph) AND (open OR accessible) AND (data OR info*) AND (sources)) AND PUBYEAR > 2014 AND (LIMIT-TO (SRCTYPE , "j"))

Como resultado se obtiene un total de 56 publicaciones relacionadas en la revista Scopus, mientras que, 23 publicaciones corresponden a Web of Science. Los resultados deben aportar con técnicas o herramientas de enriquecimiento de grafos de conocimiento.

En el siguiente apartado se seleccionan las publicaciones de mayor relevancia.

1.2.4 Resultados de revisión

Se utilizaron dos bases de datos científicas: Socopus y Web of Sicence. En la Tabla 3 se detallan los resultados de cada búsqueda, se encontraron un total de 79 artículos que probablemente contengan información relacionada con la presente investigación.

Las revistas encontradas se calificaron con los tres factores de impacto: JCR,SJR y h5 de Google. Seguidamente se analizaron los objetivos y el alcance de cada publicación, para filtrar a todos los trabajos que mayor aporte tengan para esta nueva investigación.

Tabla 4

Resultados de búsqueda

Bases de datos	Número de artículos	Número de artículos seleccionados
Socups	56	18
Web Of Science	23	6

1.2.5 Análisis de resultados

Los trabajos analizados abarcan temáticas importantes al dominio de la investigación, que es el enriquecimiento de grafos de conocimiento, en esta sección se realiza la descripción principal de cada trabajo y se exponen las técnicas, dominios, modelos y herramientas para el cumplimiento de los objetivos.

En el trabajo realizado por (Gharibi, Zachariah, & Rao, 2020) denominado “FoodKG: A Tool to Enrich Knowledge Graphs Using Machine Learning Techniques” se propone una herramienta de software novedosa para enriquecer un grafo de conocimiento utilizando aprendizaje automático, la metodología empleada se basa en cuatro etapas.

La primera etapa del modelo planteado delimitar el alcance de la solución a un vocabulario denominado AGROVOC. La segunda fase denominada extracción de entidades tiene como objetivo unificar relaciones y entidades para facilitar el enriquecimiento. La tercera fase introduce el término de similitud semántica haciendo uso de algoritmos propuestos para medir la similitud de dos vectores. En la última etapa incorpora datos estructurados para el enriquecimiento utilizando técnicas de predicción de enlaces como: Specialization Tensor Model (STM), el cual es un modelo neuronal de retroalimentación para clasificar relaciones léxico-semánticas para pares de palabras.

Por otra parte (Abu-Salih et al., 2021) tiene como objetivo enriquecer un grafo de conocimiento bajo el dominio política, y señala que la información base para enriquecer un grafo de conocimiento se procesa bajo el uso del lenguaje de mapeo RDF (RML) con el fin

de estandarizar los datos colectados. Además, se aborda las herramientas utilizadas para la incrustación de grafos como: Python y tensorFlow. El modelo planteado consta de cuatro fases, el primer paso consiste en seleccionar fuentes de datos, para la investigación se hacen uso de twitterr, wikipeedia y revistas de noticias. En el segundo paso se selecciona las herramientas que soporta el modelo planteado. En tercera fase se realiza la extracción de entidades y clasificaciones de dominio. En la última etapa se aborda la predicción de enlaces, clustering, clasificación y visualización.

Además (Tempelmeier & Demidova, 2021) menciona que a menudo los grafos de conocimiento como Wikidata y DBpedia están incompletos y que existen otras fuentes de datos que pueden complementar la información; sin embargo, el descubrimiento de enlaces es desafiante debido a: la falta de esquemas exactos y la heterogeniedad en las representaciones. El trabajo propone un enfoque para predecir enlaces denominado OSM2KG el cual es un modelo supervisado que captura similitudes semánticas en una incrustación. El dominio trabajado corresponde a OSM (Open Street Map), en el modelo propuesto se hace uso de ConvE que tiene como objetivo incrustar relaciones en una matriz bidimensional.

Otra propuesta presentada por (Saeed, Chelmis, & Prasanna, 2019) se enfoca en desarrollar un método escalable basado en caminos aleatorios bidireccionales, el algoritmo utilizado emplea esquemas de ponderación con el fin de generar incrustaciones a los gráficos. Para la incrustación de conocimiento en el grafo se trabaja bajo RDF. El autor aborda problemas latentes como la similitud semántica, y se limita al cálculo de dos entidades al mismo tipo.

Por otra parte (Cao, Wang, Huang, & Hu, 2020) exponen los retos a los que se enfrentan los grafos de conocimiento, como la información desactualizada y la carencia de hechos. En la actualidad enriquecer un grafo de conocimiento se basa en completar enlaces faltantes o encontrar información; sin embargo, recalca que investigaciones abordan parte de los problemas de los grafos de conocimiento, es por ello que el modelo propuesto por el autor realiza tareas de predicción de propiedades faltantes e infiere hechos, el enfoque

consiste en comparar entidades populares similares, con el fin de encontrar hechos faltantes, para la predicción de enlaces hace uso de una red neuronal gráfica GNN y para inferir los hechos hace uso de modelos probabilísticos

Finalmente, en base al aumento de investigaciones para el enriquecimiento de grafos, compañías como Amazon distribuyen paquetes escalables de software que simplifica el proceso de incrustación de gráficos, denominado DGL-KE con el fin de abordar a grafos de conocimiento de gran escala, aplicando tareas de predicción de enlaces e inferencia de similitud de incrustaciones.

Además de analizar las investigaciones realizadas para enriquecer los grafos de conocimiento, se revisó la literatura para encontrar los desafíos principales de los grafos. el autor (Petzold et al., n.d.) señala que el principal desafío es el rápido crecimiento de la información en la web dando como resultado datos incompletos y que la mayoría de los científicos e industrias trabajan sobre información relacional aumentando los silos de información.

En base al análisis realizado en esta sección, se procede con una revisión a detalle de los modelos y algoritmos para el enriquecimiento de grafos de conocimiento, con el fin de retroalimentar la información de las investigaciones.

Los enfoques propuestos por la mayoría de los autores se inclinan a la incrustación de grafos como parte del enriquecimiento, teniendo como base fases previas de análisis en sus modelos. La predicción de enlaces es una de las técnicas más utilizadas.

Igualmente, en los trabajos desarrollados se hacen referencia a diferentes algoritmos que se han utilizado y medido para enriquecer un grafo de conocimiento. uno de ellos es TrasnE que tiene como objetivo modelar datos multirelacionales es decir, a grafos dirigidos, en la publicación realizada por (Andreotti, Emmanuele, Fontanella, Zanier, & Luise, 2015) experimentan la incrustación de gráficos a través de bases de conocimiento como Wordnet y Firebase, el modelo se basa en extraer patrones de conectividad local o global entre entidades, el modelo mencionado maneja relaciones 1:1. Del mismo modo, métodos como TransH y TransR son versiones mejoradas, aportando el tipo de relación, en el caso de

TransH se maneja una relación 1:N mientras que TransR tarta a las relaciones por separadas y calcula la distancia entre entidades.

Otros métodos de incrustación de grafos son los denominados métodos de factorización entre ellos DistMult que constan de dos capas, una de incrustación y puntuación, el modelo consume triples del grafo de conocimiento, el cual consiste en aprender incrustaciones de un objetivo bilineal (Yang, Yih, He, Gao, & Deng, 2015), de lo contrario el método ComplEx creado por (Trouillon, Welbl, Riedel, Ciaussier, & Bouchard, 2016) hace uso de incrustaciones complejas en base a la matriz Hermitian, en el uso de los diferentes algoritmos tiene como fin de comprender automáticamente la estructura de grandes bases de conocimiento.

Los métodos basados en CNN o tipo de modelo de red neuronal, entre los principales ConvE es un modelo de red convolucional 2D multicapa para la predicción de enlaces, el modelo fue propuesto por (Dettmers, Minervini, Stenetorp, & Riedel, 2018) detalla que las interacciones entre entidades de entrada y las relaciones son modeladas por capas convolucionales, en el modelo expuesto se analizan todas las entidades silmultaneamente. Otro modelo basado en CNN es CapsE que representa cada triple como una matriz de tres columnas, utiliza la red de capsulas para modelar las tripletas seguidamente de una capa convolucional para la predicción de enlaces.

Con base al análisis realizado en los diferentes trabajos relacionados, se responde a las preguntas de investigación planteadas, la primera pregunta menciona lo siguiente: ¿Cuáles son los desafíos actuales para enriquecer un grafo de conocimiento? La inferencia del conocimiento, la predicción de enlaces, la precisión en las fuentes de información y los silos de información son los principales desafíos de los grafos de conocimientos.

La segunda pregunta de investigación menciona lo siguiente ¿Qué criterios o técnicas existen para enriquecer un grafo de conocimiento? Para enriquecer un grafo de conocimiento se debe tener claro que existen varios tipos de enriquecimiento como: predecir enlaces faltantes dentro de un grafo de conocimiento o completar un grafo de conocimiento haciendo referencia a otras fuentes de datos.

La tercera pegunta de investigación menciona lo siguiente ¿Cuáles es el grafo de conocimiento abierto más relevante en la web? Se debe tener en cuenta que la calidad de los datos debe prevalecer en un grafo de conocimiento, entre las principales fuentes de datos abiertos constan: DBpedia, Freebase, OpenCyc, Wikidata y Yago; sin embargo, en el análisis realizado por (Bartscherer, Menne, & Rettinger, 2017) demuestran que en base 34 criterios de calidad pueden ser analizados, en el estudio realizado Wikidata obtiene la mayor puntuación seguido de DBpedia.

Por último, la cuarta pregunta de investigación ¿Cuáles son los dominios que han sido trabajados para enriquecer un KG? Dentro de los trabajos relacionados, los estudios se han enfocado en trabajar bajo un dominio, los principales están enfocados, en la medicina y la política.

Conclusiones

En el presente capítulo se describen las principales investigaciones actuales referentes al enriquecimiento de los grafos de conocimiento, teniendo como resultado investigaciones significativas en el dominio.

El principal reto de las investigaciones es generar conocimiento a partir de los datos; sin embargo, muchas de las investigaciones se centran en un solo dominio. El presente trabajo busca enriquecer un grafo de conocimiento RDF con una fuente de datos abiertos, estableciendo mecanismos utilizados en trabajos relacionados

Capítulo dos

Marco Teórico

En el presente capítulo se describen los conceptos bases para el entendimiento de la investigación, a continuación, se detalla cada una de las secciones del capítulo: **1.1 La Web Semántica:** Se realiza una introducción a la Web Semántica, conjuntamente se hace referencia a la arquitectura y características de los datos abiertos.

1.2 Datos abiertos: Dentro de esta sección se menciona la importancia del uso de los datos abiertos.

1.3 Grafos de conocimientos: Se profundizan conceptos relevantes y se conceptualiza las principales fuentes de grafos de conocimientos abiertos.

1.4 Técnicas de incrustación de grafos: Se compara las técnicas de predicción de enlaces existentes.

1.5 Herramientas Se detallan las herramientas, frameworks, API´s, que ayudan a generar contenido RDF y vocabularios actuales.

2.1 Web Semántica

La Web Semántica proporciona en la Web datos definidos y enlazados, permitiendo que aplicaciones heterogéneas integren y reutilicen la información presente en la Web (Martínez Arellano & Amaya Ramírez, 2017), la filosofía resaltada por muchos de los autores se resume en cuatro características importantes:

- Implementación de modelos de metadatos para describir recursos de información.
- Uso de vocabularios RDF para representar metadatos.
- Desarrollo de esquemas RDF y ontologías para describir relaciones entre los recursos.
- Interconexión y reutilización de fuentes de datos RDF, que permite su integración mediante procesos automáticos.

Dentro de la Web Semántica existen anotaciones RDF y RDFs que facilitan la representación de los datos encontrados en la Web. Por una parte, RDF es el marco para representar la información en la Web en sus diferentes serializaciones como: Ntriples, JSONLD, Rdfa y Turtle. Cada una la representación corresponde a un formato para modelar RDF, mientras que RDFs proporciona un vocabulario que proporciona semántica a los datos

Hoy en día existen varias formas de representar el conocimiento. La Web Semántica establece un conjunto de vocabularios de modo que facilite el intercambio de los metadatos.

2.2 Ontologías

(Tello, 2020) Asegura que las ontologías son un soporte para la Web Semántica, definida por conceptos y relaciones de algún dominio, para dar sentido a la Web Semántica se necesita representar el conocimiento de forma legible para las máquinas, el autor menciona seis componentes claves para formalizar ontologías: clases, atributos, relaciones, funciones, axiomas e instancias:

- **Clases o Conceptos:** Son ideas que se intentan formalizar; además, un concepto puede ser algo sobre lo que se dice.
- **Atributos:** Los atributos representan la estructura interna de los conceptos, es decir propiedades que describen los conceptos.
- **Relaciones:** Son enlaces entre conceptos de un mismo dominio.
- **Instancias:** Se las utiliza para representar elementos de un determinado concepto. **- Axiomas:** Son expresiones que siempre son verdaderas y que se declaran sobre relaciones que deben cumplir los elementos de la ontología.

2.3 Datos abiertos

(Murray-Rust, 2008) considera que los datos abiertos son un término emergente en el proceso de definir como los datos pueden ser publicados y reutilizados sin barreras de precio o permisos, el autor enfatiza que la reutilización constituye el valor principal de los

datos abiertos. Hoy en día, los principales productores de datos abiertos son los gobiernos, la ciencia y grandes organizaciones internacionales.

Para que los datos puedan determinarse como “abiertos” es necesario que cumplan varios principios, (Mauthner & Parry, 2013) señala que la accesibilidad, usabilidad, calidad son los principios fundamentales que deben poseer los datos abiertos. La reutilización, redistribución e integración son factores clave de los datos abiertos con otros conjuntos de datos.

2.4 Grafos de conocimiento

Son modelos de dominio de conocimiento en una estructura de grafo, dentro de los grafos se incluye información semántica. La estructura de un grafo de conocimiento son los nodos y aristas, cada una de ellas almacena información de entidades y relaciones. (Manzano, 2015) lo define como una red semántica y una base de conocimiento con estructura gráfica que define entidades y relaciones, menciona que los grafos de conocimiento han cambiado el método tradicional de recuperar la información.

Los grafos de conocimientos integran información en una ontología y aplican un razonador para derivar nuevos conocimientos, (Saorín, 2019) señala numerosas ventajas en referencia a los grafos de conocimiento, a continuación se detallan las características principales según el autor:

- Rendimiento en comparación de realizar consultas en una base de datos SQL.
- Alta escalabilidad a grandes volúmenes de datos.
- Mayor capacidad y rapidez de procesamiento de consultas.
- Esquema flexible para la incorporación de conocimiento.
- Capacidad de inferencia para descubrir nuevo conocimiento.

2.5 Arquitectura de grafos de conocimiento

Para el desarrollo de un KG (Zhao, Han, & So, 2018) sugiere tres fases: extracción de conocimiento, construcción de conocimiento y gestión del conocimiento. Para crear el conocimiento en un KG basado en ontología existen dos enfoques principales: ascendente y descendente, en el primer enfoque extrae las instancias de conocimiento de los datos abiertos, mientras que el segundo, se debe definir en un inicio la ontología, el esquema y seguidamente las instancias.

Para empezar en la construcción de un KG, las fuentes principales de conocimiento incluyen datos estructurados, semiestructurados y no estructurados. Actualmente, existen herramientas para la extracción del conocimiento en entidades y relaciones, mientras que, la construcción del conocimiento se basa en crear una ontología alineada a la evaluación de calidad. En esta etapa se realiza la estandarización de los datos, trabajando en la coincidencia común e inferencia de datos, del mismo modo, para el almacenamiento del grafo de conocimiento existen dos alternativas: Basado en RDF y bases de datos de grafos.

Actualmente, existen dos formas respaldadas de los principales proveedores del mercado para representar el conocimiento en grafos: Basados en RDF y Grafos de propiedades etiquetadas.

2.5.1.1 Grafos de conocimientos basados en RDF

(Arenas, Cuenca Grau, Kharlamov, Marciuška, & Zheleznyakov, 2016) destaca que muchos grafos de conocimiento existentes están disponibles en la Web y se pueden exportar como conjuntos de datos RDF. Señala que para transformar datos heterogéneos abiertos como JSON, CSV y XML a datos RDF se utiliza el lenguaje de mapeo RML, definiendo entidades y propiedades.

Los KG RDF son un modelo que se presenta como la web semántica que comprende datos enlazados.

(Arnaout & Elbassuoni, 2018) lo define como un conjunto de triples RDF multiple etiquetado, donde las etiquetas de nodo son URI que hacen referencia a los recursos o literales y las etiquetas de borde corresponde a los predicados.

2.5.1.2 Grafos de conocimientos basados en propiedades. (Angles, 2018) lo define como un multigrafo dirigido y etiquetado, la principal característica de que cada nodo o borde pueden ser un nodo vacío, cada nodo representa una entidad, los bordes presentan las relaciones y una propiedad representa una característica especifica de una entidad o relación.

2.5.2 Bases de datos orientadas a RDF

Para almacenar el conocimiento existen sistemas como: Jena2, 3store, RDFStore, 4Store, RDF3X y Virtuoso, que proporcionan características de un lenguaje de consultas SPARQL, entre las más populares se encuentran:

- **Apache Jena**

  Es un framework de código abierto escrito en Java enfocado para crear aplicaciones de Web Semántica, su principal característica es un API para leer, procesar y escribir ontologías, que proporciona un soporte para OWL, entre los lenguajes se admiten SPARQL y RDQL, además, los grafos de conocimiento se representan bajo un modelo abstracto.

- **RDF Store JS**

  Es una librería de JavaScript para almacenar grafos RDF, con un soporte para el manejo de datos y consultas SPARQL, al ser escrito en javascipt se lo puede ejecutar desde un navegador web. La librería actualmente dispone de un API de eventos experimental para los grafos RDF, el cual permite observar cambios y recibir notificaciones cuando cambia parte del grafo.

- **Virtuoso RDF**

  Es un servidor de datos que gestiona los datos RDF que admite diferentes serializaciones como: NTriples, XML y RDF, que son compatibles con el lenguaje de consulta SPARQL. En su modelo no existe un límite de grafos que se puedan almacenar. Virtuoso RDF incluye varias funciones para desarrolladores como analizar grafos, inserciones y mapeo de relaciones. En la Tabla 5 se resumen las principales características de cada herramienta que se utiliza para el almacenamiento en RDF:

Tabla 5

Almacenamiento basado en RDF

	ALMACENAMIENTO BASADO EN RDF					
Herramienta-Framework	**Lenguajes soportados**	**API de interfaces**	**Api de eventos**	**Implementación**	**Versión estable**	**Mantiene Soporte**
Apache Jena	RDQL, SPARQL	X		Java		Sí
RDF Store JS	SPARQL	X	X	JavaScript		Sí
Virtuoso RDF	SPARQL, SPASQL, SQL	X	X			Sí

2.5.3 Bases de datos orientadas a grafos (BDOG)

Por otra parte, las bases de datos de grafos son otra forma de almacenar conocimiento, incluye: nodos, bordes y propiedades de los grafos. La ventaja principal que indica (Zhao et al., 2018) es la cantidad de algoritmos de minería de grafos que existen inmersos en estas herramientas; no obstante, los problemas de gestión de datos son crecientes, y a ello se suma el alto costo de mantenimiento.

Las bases de datos más populares orientadas a grafos se caracterizan por representar la información en vértices y aristas. Las BDOG pertenecen al grupo No SQL, el principal beneficio es el alto rendimiento de búsqueda de resultados, entre las más relevantes existen:

- **Neo4j**

  Es la plataforma de grafos líder en el mundo, está diseñada para la gestión, almacenamiento y consultas optimas de nodos y relaciones. Para el acceso a los datos se utiliza el lenguaje de consulta Cypher el cual permite a los usuarios finales recuperar y almacenar los datos.

  Neo4j ofrece un complemento que permite el manejo de RDF y vocabularios asociados, entre principales funcionalidades incluye:

  - Integración con componentes que utilicen RDF

- Importación y exportación de RDF en sus diferentes seralizaciones.
- Validación de grafos.

Además, brinda un servicio en la nube totalmente gestionado.

- **ArangoDB**

  Es una base de datos de código abierto multi-modelo, dispone de un API para administrar la base de datos, su principal característica es el uso de un solo lenguaje, permitiendo realizar consultas entre diferentes modelos como: clave-valor, documentos y grafos. El Servicio gestionado se adquiere desde azure y tiene una interfaz web para su administración.

- **Amazon Neptune.**

  Es una base de datos gráfica de código abierto, totalmente administrable que facilita la creación de aplicaciones que funcionan con conjuntos de datos conectados, en donde se almacenan millones de relaciones, su eje principal es la rapidez de consultas, también admite APIs a grafos abiertos para SPARQL.

2.6 Técnicas de incrustación de grafos de conocimientos

Las técnicas de incrustación se pueden utilizar para afinar el conocimiento de un grafo, permitiendo predecir información faltante. (Mittal & Choudhary, 2014) define a la incrustación de grafos de conocimiento como un enfoque para transformar los grafos de en espacios vectoriales, dividiéndolos en dos grandes grupos: modelos de traducción y modelos de coincidencia semántica.

Existen dos enfoques de enriquecimiento de grafos de conocimiento: la predicción de enlaces y la clasificación de triples, en el siguiente apartado se detallan cada una de ellas:

2.6.1.1 Predicción de enlaces

Este método se caracteriza por encontrar una entidad que pueda ser representada como un hecho (h,r,?) o (?,r,t), según (Rossi, Barbosa, Firmani, Matinata, & Merialdo, 2021)

la predicción de enlaces es una forma de extender un grafo de conocimiento, deduciendo información faltante entre entidades.

Por otra parte (Muhan Zhang, 2018) enfatiza que la predicción de enlaces proporciona que dos nodos de una red tengan un enlace, el enfoque analizado por el autor son los métodos heurísticos que calculan las puntuaciones de similitud entre los nodos.

2.6.1.2 Clasificación de triples. De acuerdo a (Rossi et al., 2021) el enfoque se basa en identificar si un triple dado es el correcto, tiene como objetivo retornar una respuesta verdadera o falsa utilizando una función de puntuación para calcular triples similares, si la puntuación es mayor al umbral configurado se considera que el hecho es un triple erróneo.

2.7 Grafos de conocimientos abiertos

Los grafos de conocimiento se originan de la necesidad de modelar los datos conectados, la principal característica es el enriquecimiento con otros datos, entre los principales grafos de conocimiento existen: DBpedia, Freebase, OpenCyc, Wikidata, y YAGO. (Bartscherer et al., 2017) señala que existen varios factores al momento de hacer uso de un grafo de conocimiento, ya sea en la industria, investigación o educación, el autor realiza una revisión profunda en base a 34 criterios de calidad que los grafos son expuestos, a continuación, se presentan cada uno de ellos:

- **DBpedia**

  Es un proyecto que realiza la extracción de datos desde Wikipedia, permitiendo a los usuarios consultar semánticamente las relaciones, propiedades de Wikipedia, API y su punto final de consulta que es un SPARQL.

- **Freebase**

  Es una base de datos colaborativa, la cual está compuesta por metadatos, los datos son totalmente libres, su principal objetivo es crear un recurso global para que personas y maquinas tengan acceso a la información. El API no se

encuentra disponible; sin embargo, google proporciona un volcado de datos que contiene 1.900 millones de triples con cohorte al 2020.

- **OpenCyc**

  Se trata de un proyecto de inteligencia artificial, que contiene millones de términos y afirmaciones. El grafo provee un API para el acceso a la base de conocimiento donde se puede navegar, consumir y editar.

- **Wikidata**

  Es una base de conocimiento colaborativa, cuyo objetivo principal es proporcionar una fuente común de datos, para ser utilizados en proyectos, e incluye un API para el acceso a los datos.

- **Yago**

  La base de conocimiento semántica es derivación de Wikipedia y otras fuentes de datos, está compuesta por 17 millones de entidades y contiene más de 150 millones de datos sobre las entidades.

En base a la evaluación realizada por el autor (Heiko Paulheim, 2016) referente a los grafos de conocimiento, indica que existen nueve fuentes populares de grafos de conocimiento en la web, en la Tabla 6 se describe cada una de los KG, haciendo énfasis en el número de instancias, hechos, tipos y relaciones de los grafos de conocimiento.

Tabla 6

Grafos de conocimiento en la Web

Nombre	Instancias	Hechos	Tipos	Relaciones
DBpedia (English)	4,806,150	176,043,129	735	2,813
YAGO	4,595,906	25,946,870	488,469	77
Freebase	49,947,845	3,041,722,635	26,507	37,781
Wikidata	15,602,060	65,993,797	23,157	1,673

Nota: Tomado de (Heiko Paulheim, 2016)

2.8 Comentarios finales

El eje principal de la investigación es enriquecer grafos de conocimiento desde fuente de datos abiertos. Como resultado final del Capítulo II se definen los conceptos claves de la presente investigación, se toman en cuenta: herramientas, mecanismos y datos interesantes que ayudan al desarrollo del objetivo planteado que consiste en construir un modelo para el enriquecimiento de grafos de conocimiento RDF.

Capítulo tres

Planteamiento del problema

El capítulo se divide en cinco secciones detalladas a continuación:

3.1 Contexto: En esta sección se contextualiza la investigación y el enfoque planteado de la investigación.

3.2 Planteamiento del problema: Se detalla el problema actual, tomando en cuenta el Capítulo I de trabajos relacionados.

3.3 Objetivos: Se describen el objetivo general y objetivos específicos que se plantearon en la investigación.

3.4 Entregables: Se describe la propuesta y la arquitectura en base al alcance y objetivos planteados, del mismo modo se describen los componentes arquitectónicos del modelo.

3.1 Contexto

Frente al aumento de disponibilidad de datos vinculados han dado cabida a los grafos de conocimiento a gran escala en todo el mundo, se han convertido en un componente cada vez más crucial en los sistemas de inteligencia artificial, potenciando asistentes digitales e inspirando varios proyectos de transformación digital a gran escala, incrementando la investigación de técnicas avanzadas de extracción y enriquecimiento de información.

Frente al auge de utilizar grafos de conocimiento existen fuentes de datos abiertos que pueden ser de gran aporte para mantener la información conectada.

3.2 Problema

Estructurar con precisión el conocimiento en un KG involucra asegurar la calidad sintáctica y semántica de la información; además, de abordar un enriquecimiento racional en los grafos de conocimiento. Es necesario hacer uso de procesos automáticos o semiautomáticos que ayuden a gestionar de manera adecuada todo el proceso de enriquecimiento de un KG.

Sin embargo, “enriquecer” o “generar conocimiento” en un KG es una tarea compleja, el hecho de que un grafo de conocimiento este enriquecido semánticamente, implica que hay un significado asociado a las entidades en el grafo, teniendo como base una ontología.

El problema actual es la carencia de vínculos o referencias a fuentes de datos abiertos, produciendo silos de información que no aportan ningún valor a los datos.

En este contexto, la investigación pretende enriquecer grafos de conocimiento, conectando la información con fuentes de datos abiertos, analizando lo enfoques existentes de enriquecimiento de grafos, con el fin de evaluar y optimizar estudios ya realizados, el resultado final es aportar un marco de trabajo para enriquecer cualquier grafo de conocimiento RDF.

En base a expuesto en secciones anteriores, se plantea construir el modelo de enriquecimiento de grafos desde Wikipedia como fuente de datos abiertos. En la propuesta se plantea que los grafos de conocimiento se enriquezcan agregando nuevas relaciones de fuentes de datos externas

3.3 Objetivos

3.3.1 Objetivo General

Desarrollar un Método de enriquecimiento de Grafos de Conocimiento RDF desde fuentes de datos abiertos

3.3.2 Objetivos Específicos

- Colectar y pre-procesar información relacionada con enriquecimiento automático de Grafos de Conocimiento.
- Diseñar una arquitectura para generar nodos candidatos
- Desarrolla un prototipo sobre el método de enriquecimiento
- Diseñar casos de uso de enriquecimiento de KG desde fuentes de datos abiertas
- Medir el rendimiento del marco propuesto a través de casos de uso sobre el prototipo desarrollado

3.4 Entregables

Los resultados esperados en el Trabajo de fin de master es cumplir con los siguientes ítems:

- Investigación de trabajos relacionados que permita plantear una solución viable para el enriquecimiento de grafos.
- Diseño de un Método de enriquecimiento de Grafos de Conocimiento basado en la desde fuentes de datos abiertos.
- Desarrollo de un prototipo capaz de generar nodos candidatos desde fuentes de datos abiertas,
- Planteamiento de casos de uso para medir el rendimiento de la solución planteada

Capítulo cuatro

Propuesta

En el siguiente capítulo se presenta un marco para enriquecer grafos de conocimiento, bajo cualquier dominio, la tarea principal es generar nodos candidatos para plegarlo a un nodo existente, la propuesta hará uso de procesamiento de lenguaje natural para abordar el enriquecimiento semántico, en conjunto con herramientas de extracción de conocimiento para obtener entidades o nodos. Además, se desarrollará un prototipo bajo la arquitectura propuesta.

4.1 Propuesta de la solución

Para cumplir con el objetivo planteado de realizar un modelo de enriquecimiento de grafos de conocimiento desde fuentes de datos abiertos. Como primera fase se realizó una revisión previa de investigaciones relacionadas que describen pautas a considerar en sus enfoques planteados.

Para el enriquecimiento de grafos se hará uso de Wikipedia como fuente de datos abiertos, debido a la gran mayoría de instancias alojadas en el gafo de conocimiento descrito en el Capítulo 1.

El modelo de enriquecimiento usa el procesamiento de lenguaje natural para detectar el tipo de entidades o nodos que se van a enriquecer, se plantea el uso de Google Cloud Natural Language como la principal herramienta. El objetico principal es generar metadatos necesarios para enriquecer el grafo de conocimiento.

A partir de la metadata, se construyen consultas SPARQL, para obtener información relacionada al nodo a enriquecer, se realiza una conexión directa con el servicio de fuente de datos abiertos Wikipedia como fuente principal de conocimiento.

El modelo planteado permitirá medir el número de nodos iniciales y el número de nodos enriquecidos, realizando una validación de los nuevos datos plegados al grafo de conocimiento inicial.

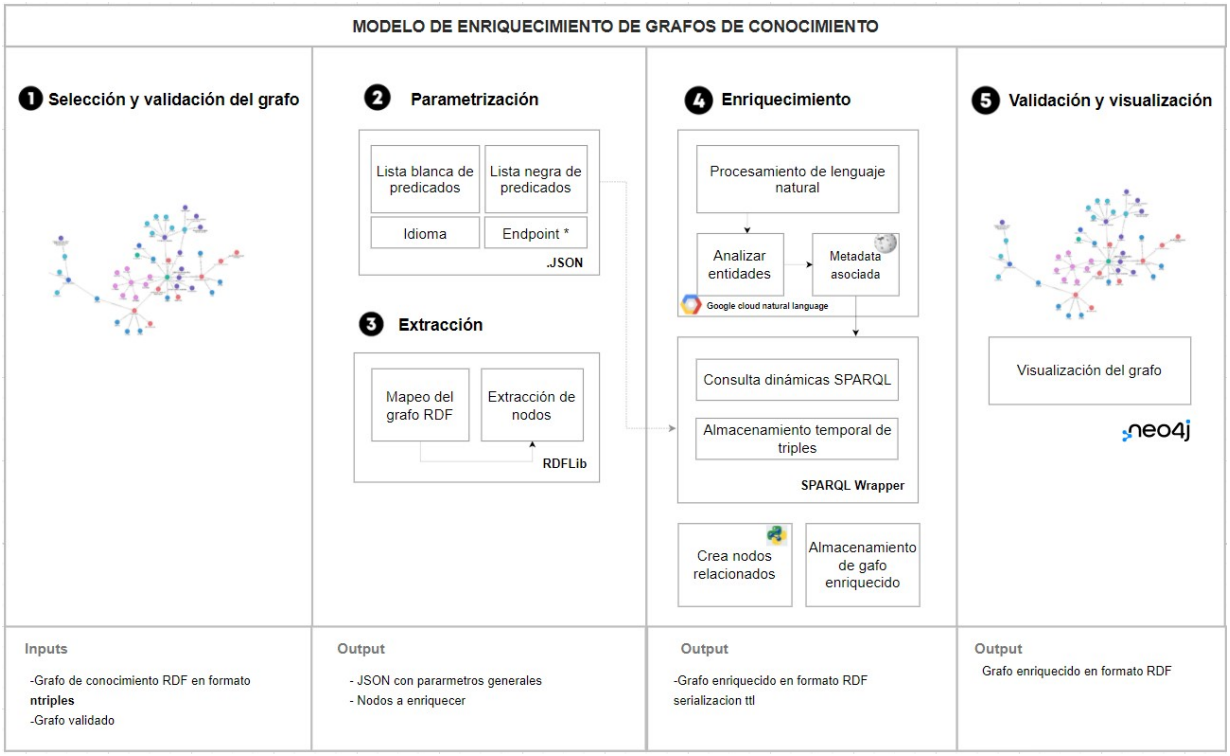
4.1.1 Componentes arquitectónicos

Para lograr los objetivos planteados en el presente trabajo, es necesario construir un modelo sólido para el enriquecimiento de grafos desde fuentes de datos abiertos, la misma está enfocada en utilizar frameworks que faciliten el manejo de los grafos de conocimiento y agilicen el proceso de reconocimiento de entidades.

En la Figura 2 se muestra el modelo propuesto para enriquecer un grafo de conocimiento, el enfoque consta de cuatro fases:

Figura 2

Método de enriquecimiento de grafos



La solución se construye bajo el lenguaje de programación Python el cual permite el uso de múltiples bibliotecas enfocadas en el manejo de grafos RDF, en el siguiente apartado se describen cada una de las fases del modelo planteado:

- Fase I **Selección y validación del grafo**:

El modelo tiene como primer requisito partir de un grafo existente, que inicia de un archivo en formato RDF (Marco de descripción de recursos) y puede ser representado en la serializacion N-Triples.

La biblioteca a utilizar es rdflib que permite realizar la validación del grafo en tiempo real.

- Fase II **Parametrización:**

  Se configura parámetros relevantes para el enriquecimiento del gafo, parametrizando valores como: limite, idioma, profundidad, lista blanca y negra de los predicados y el edpoint, este último ítem busca que futuras investigaciones se expanda las consultas a diferentes fuentes de datos abiertas como dbpedia.

- Fase III **Extracción de nodos:**

  Para el enriquecimiento del grafo de conocimiento se itera por cada una de los triples y se extrae el recurso o el nombre de la entidad. Esta fase es esencial ya que es la base del enriquecimiento, para la extracción de nodos se usa la biblioteca rdflib construida en Python para iterar sobre el rdf, la biblioteca rdflib permite realizar serealizaciones RDF/XML.

- Fase IV **Enriquecimiento:**

  Como parámetros de entrada se usa el resultado de la Fase II y de los nodos extraídos en la Fase III, a continuación, se describe cada una de las sub fases:

  - **Análisis de entidades o nodos:**

    En esta fase se usa el lenguaje de natural de google cloud, que permite extraer la entidad asociada de una palabra, la búsqueda de la entidad o recurso retorna los metadatos asociados.

    El objetivo de esta fase es utilizar mecanismos fiables que permitan agilizar el desarrollo y descubrimiento de conocimiento. Actualmente google ofrece el procesamiento de lenguaje natural en cloud

permitiendo acceder a un API para recolectar información asociada de un texto origen, los datos que proporciona google son: el tipo entidad, enlace de wikipedia y el identificador único del grafo de google.

Como resultado de la subfase se obtienen los enlaces de wikipedia de los nodos encontrados en wikidata. ○ **Conexión a fuentes de datos abiertos:**

Con los metadatos obtenidos en la subfase anterior, se construye consultas asociadas al recurso de wikidata, esta fase permitir ejecutar consultas SPARQL de forma remota al endpoint de wikidata, además de almacenar los triples temporalmente para realizar una iteración de cada nodo e ir plegando la coincidencia de cada triple.

- Fase V **Validación y visualización:** Esta última fase permite evaluar el rendimiento del modelo planteado, se obtiene el número de nodos iniciales vs los nodos enriquecidos; así mismo, se muestra el resultado de enlaces de wikidata encontrados, agregando los resultados en el archivo del grafo final.

Como resultado de la sección 3.4.1 se obtiene el diseño de la propuesta de enriquecimiento de grafos, en donde se ha descrito el flujo del modelo y las diferentes herramientas utilizadas en cada fase.

4.1.2 Descripción de herramientas

En la presente sección se describen las herramientas, técnicas y mecanismos utilizados en el prototipo a desarrollar:

4.1.2.1 Python. Python es un lenguaje de alto nivel que tiene una extensa colección de bibliotecas y frameworks, es muy popular en la ciencia de datos ya que permite manejar funciones matemáticas y científicas, además de poseer una comunidad solida del código abierto.

La mayoría de los trabajos relacionados descritos en el Capito II hacen uso de python como el lenguaje de programación base para iterar sobre grafos de conocimiento. En el presente trabajo se analizó las bibliotecas actuales que permitan el manejo de grafos

RDF.

4.1.2.2 RDFLib. Es una biblioteca de Python que permite trabajar con formatos grafo RDF, la biblioteca se caracteriza por almacenar triples en memoria en las diferentes serializacion RDF, además permite fusionar grafos simples en su versión más reciente. Las propiedades clave para itera sobre un grafo son:

- **URIRef:** Permite referenciar una URI dentro del grafo RDF.
- **BNode:** Permite crear nodos en blanco, como sujeto o un objeto dentro de las representaciones
- **Literal:** Permite agregar valores de atributos en RDF.

La biblioteca RDFLib permite iterar sobre el grafo entre las principales operaciones se contemplan los siguientes:

- **Agregar triples:** Esta función permite agregar un triple usando los nodos y nombres definidos previamente, para agregar un nodo con rdflib se debe de colocar la sentencia ***g.add(({sujeto}, {predicado}, {objeto}))*** cada uno de los parámetros debe contener una propiedad descritas en el punto anterior.
- **Eliminar triples:** Del mismo modo para eliminar un nodo se debe de referenciar las tres claves del triple como es el sujeto, predicado y objeto, colocando la siguiente sentencia ***g.remove(({sujeto}, {pedicado}, {objeto}))***

4.1.2.3 Google Cloud Natual Language. Proporciona a desarrolladores una forma de analizar el lenguaje natural, partiendo de análisis de entidades, sintaxis, opiniones, emociones y clasificación de contenido, en la investigación.

Se hace uso del módulo denominado ***analyzeEntities***, el modulo permite encontrar los tipos de entidades conocidas, las principales entidades son (nombres propios, figuras públicas, lugares, referencia, etc.) por cada análisis realizado de obtienen metadatos de cada palabra, los metadatos que se usaran es la URL de Wikipedia y el tipo de entidad. Para hacer uso del módulo se debe colocar la siguiente sentencia ***client.analyze_entities.***

Un aspecto importante a tener en consideración es el idioma, ya que es el punto clave al momento de utilizar estos módulos, el módulo en mención tiene soporte para 11 lenguajes incluidos español e inglés, los cuales serán utilizados en el prototipo.

4.1.2.4 SPARQLWrapper. En un contenedor de Python que sirve para ejecutar consultas SPARQL de manera remota, para hacer uso de esta biblioteca se debe de definir el endpoint colocando la siguiente línea de código SPARQLWrapper ("https://query.wikidata.org/sparql") en el ejemplo se muestra el endpoint utilizado para el enriquecimiento como es wikidata, a partir de esta declaración se construye la consulta SPARQL.

La biblioteca tiene soporte para diferentes formatos como: JSON, XML, N3, RDF, CSV, TSV, en el caso práctico de utilizar wikidata, los datos deben ser retornados en formatos JSON, para su posterior uso.

Como resultado de este capítulo se ha diseñado la propuesta que cubre el proceso de enriquecimiento de grafos, teniendo en cuenta el alcance del TFM. Se ha descrito el flujo de la aplicación y las diferentes herramientas y Apis que serán de ayuda para el cumplimiento de los objetivos.

Capítulo cinco

Desarrollo

En el presente capítulo se detalla la implementación de la propuesta del problema planteado en el Capítulo 3, donde se describen tres componentes. Del mismo modo se adentra en las configuraciones base para que el modelo planteado soporte el enriquecimiento de grafos con las herramientas seleccionadas:

5.1 Configuración de ambiente: En esta sección se muestran las configuraciones o dependencias iniciales para el uso de las herramientas seleccionadas.

5.2 Desarrollo de funcionalidades: En esta sección se detalla la construcción de cada uno de las fases descritas en el Capítulo III, haciendo referencia a cada una de las funciones implementadas para el enriquecimiento.

5.3 Casos de uso: En esta sección se plantean los casos de uso que serán utilizados para el análisis y la evaluación del modelo planteado.

5.1 Configuración de ambiente

Como repositorio de código fuente se usa github, donde se ubicarán los diferentes insumos para el desarrollo del trabajo, la ruta de repositorio es la siguiente ***https://github.com/accardenas4/GraphEnrich/tree/main***.

5.1.1 *Instalación de dependencias*

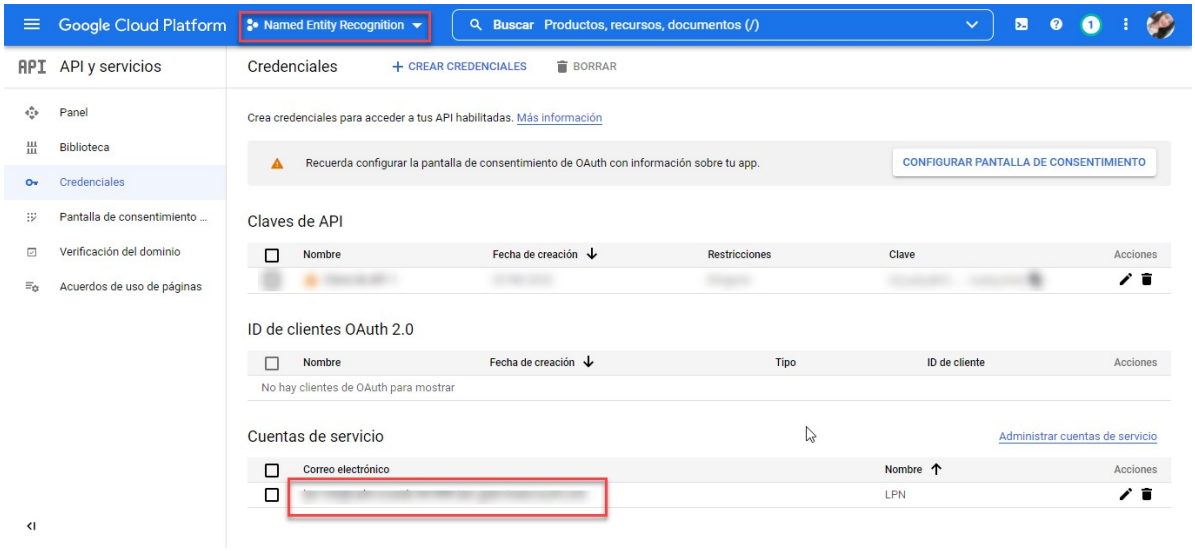
La fase inicial es la instalación de las dependencias necesarias definidas en el Capítulo 3. En el archivo **requirements.txt**, dentro del repositorio, se encuentran las 17 dependencias necesarias para iniciar con el proceso de desarrollo.

5.1.2 Habilitar API Cloud Natual Language

Se agregó un nuevo proyecto denominado ***Named Entity Recognition*** en la plataforma cloud de google en la Figura 3 se muestra el proyecto creado; así mismo las credenciales habilitadas tipo cuenta de servicio para el consumo del módulo de analizar entidades, al agregar una nueva clave se descargará un archivo .JSON que debe configurarse en las variables de entorno del computador o servidor.

Figura 3

Creación de proyecto Google Cloud



Posterior a lo mencionado se debe de parametrizar las variable **GOOGLE_APPLICATION_CREDENTIALS** y hacer referencia al archivo JSON descargado previamente.

5.2 Desarrollo de funcionalidades

Para el prototipo de enriquecimiento se crea un paquete en Python que contiene la lógica de enriquecimiento de un grafo RDF, el paquete contiene apartados esenciales que se los detalla en la siguiente sección:

- **Parametrización:** En este apartado se realizan las configuraciones iniciales, permitiendo agregar un archivo .json con variables iniciales.
- **Procesar un archivo RDF**: El objetivo es iterar cada uno de los nodos del grafo para el almacenamiento en memoria y su posterior uso.
- **Uso del Api google:** El API de lenguaje natural sirve para procesar los nodos del grafo y asociar la metadata encontrada.
- **Construcción de consultas:** El apartado de construcción de consultas dinámicas se lo trabaja bajo SPARQLWrapper, permitiendo tener tripletas listas para ser plegadas en los nodos.
- **Método de enriquecimiento** se utiliza rdflib para plegar nuevos nodos en el grafo

- **Construcción de un nuevo archivo RDF enriquecido**
- **Visualización del grafo en Neo4j**

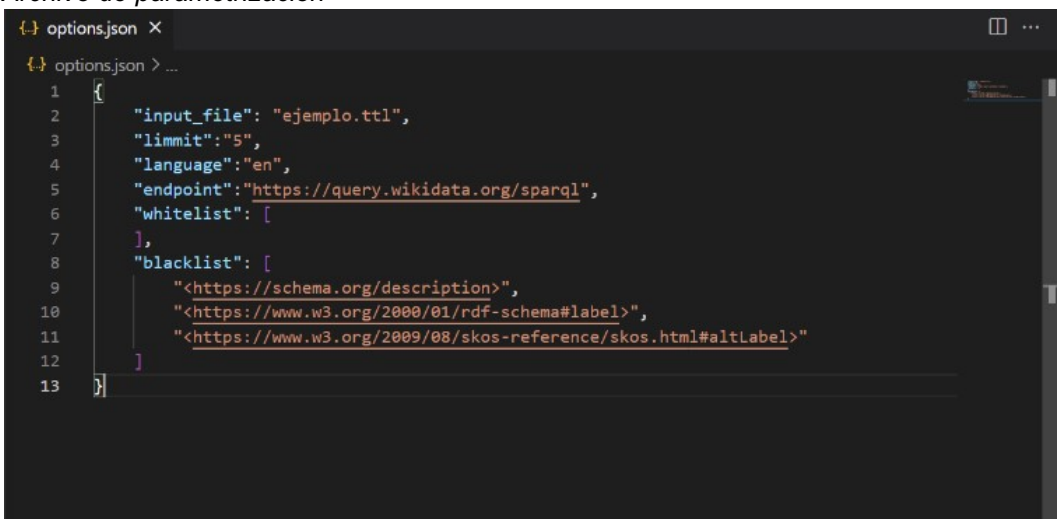
En las siguientes secciones se profundiza la implementación de cada fase propuesta en secciones anteriores.

5.2.1 Parametrización

Se crea un archivo de parametrización inicial con la extensión .json, el archivo consta de cinco variables de configuración, en la Figura 3 se muestra el fragmento de código en el proyecto:

Figura 4

Archivo de parametrización



- **Input file:** Nombre del archivo RDF que debe estar situado en la raíz del directorio del proyecto.
- **Limmit**: Limite profundidad de consultas por cada URL de wikidata desde el endpoint.
- **Languge**: Se permite la configuración del lenguaje para utilizarlo en el api de google cloud y en el wrapper SQL.
- **EndPoint**: URL del endpoint SPARQL de grafos de conocimientos abiertos, para el prototipo se usa wikidata
- **Whitelist:** Apartado para enlistar predicados únicos a enriquecer, si el parámetro no se lo completa se extrae cualquier predicado de wikidata.

- **BlackList**: Apartado para enlistar predicados que no se mostraran en los resultados de la consulta SPARQL en wikidata.

5.2.2 Procesamiento de archivo RDF

Para el procesamiento del grafo RDF, se utiliza la biblioteca rdflib, la cual permite cargar el grafo en memoria e ir iterando cada uno de los nodos, en la Figura 4 se muestra el fragmento de código que permite inicializar el grafo en base al archivo rdf y se inicializan variables para almacenar la longitud del grafo inicial, con el fin de compararlo con el grafo final, consiguiendo una métrica inicial.

Figura 5

Procesamiento de Archivo RDF

```
#Initialize the graph and load the graph
g = Graph()
g.parse(input_file)

# Data graph length
count_nodes=len(g)
count =0;

# Iterate graph for get nodes
for s, p, o in g.triples((None,  RDF.type, None)):
    #find character "/" and lenght string
    entity_lng=s.rfind('/')
    #Store entity names
    name_entity = s[entity_lng:]
```

5.2.3 *Analizar entidades*

Por cada nodo iterado se realiza una petición al Api de google Cloud Natural Language, con el fin de obtener metadatos asociados, en la Figura 6 se muestra el fragmento de código para el análisis de los parámetros definidos en la sección 4.2.1 referente a la parametrización, por cada nodo encontrado se realiza la extracción de la metadata del nodo, el metadato que se utiliza es la URL asociada a wikidata.

Figura 6

Analizador de entidades

```
# call api de google
client = language_v1.LanguageServiceClient()
# send name entity/node
text_content = name_entity
# define_type
type_ = language_v1.Document.Type.PLAIN_TEXT
# set language
language = "en"
# assemble document
document = {"content": text_content, "type_": type_, "language": language}
# set encoding
encoding_type = language_v1.EncodingType.UTF8
# analize entities
response = client.analyze_entities(
    request={'document': document, 'encoding_type': encoding_type})

# iterate from entities and metadata
for entity in response.entities:
    for metadata_name, metadata_value in entity.metadata.items():
        #get metadata from wikipedia_url
        if (metadata_name == "wikipedia_url"):
            # set URL wikidata in wd
            uri = metadata_value
            wd = '<'+metadata_value+'>'
            count = count+1
```

5.2.4 Construcción de consultas SPARQL

Por cada enlace de Wikipedia encontrado por el analizador de entidades de google se ejecuta la consulta SPARQL, la misma consta de 4 parámetros de entrada para su construcción: endpoint de Wikipedia, lista blanca de predicados, lista negra de predicados y el lenguaje. En la Figura 7 se muestra el fragmento de código que se desarrolla para cumplir con la extracción de tripletas de una fuente de datos abiertos como es wikidata.

Figura 7

Wrapper SPARQL

```
#initialize wrapper SPARQL
sparql = SPARQLWrapper("https://query.wikidata.org/sparql")
#set query to endpoint
sparql.setQuery("""
CONSTRUCT {
  ?statement  ?p ?l     }
WHERE {""" +
    wd + """ schema:about ?statement .
   ?statement ?p ?l.
   FILTER(""" +
    blacklist + """).
   FILTER(""" +
    whitelist + """)
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],"" +
    language + """. }
}limit 8
""")
#format results
sparql.setReturnFormat(JSON)
results = sparql.query().convert()
```

5.2.5 Enriquecimiento del grafo

Con base a las tripletas extraídas en la sección anterior se procede con la primera relación, el objetivo es agregar el nodo analizado al nodo origen, se hace uso del lenguaje para ontologías web OWL, la propiedad utilizada es owl:sameAs la cual se asociara al nodo inicial, para agregar la similaridad se implementa la siguiente línea de código: **g.add((s, OWL.sameAs, URIRef(sujeto))).**

Para agrega las tripletas del nuevo nodo se serializan los resultados y se agregan en base al tipo de dato, los tipos de datos soportados por la biblioteca son: URL o literales en la Figura 7 se muestra el fragmento de código para plegar nuevos nodos dependiendo del tipo encontrado en wikidata.

Figura 8

Agregar nuevos nodos

```
#validate URL or Literal
isURL = objeto.find("http")
if isURL >= 0:
    g.add((URIRef(sujeto), URIRef(predicado), URIRef(objeto)))
else:
    g.add((URIRef(sujeto), URIRef(predicado), Literal(objeto)))
```

5.2.6 Exportación de grafo enriquecido

El proceso de enriquecimiento planteado construye un archivo RDF en formato turttle, al final del archivo se muestran las métricas de enriquecimiento detallando lo siguiente: el número de nodos iniciales, número de nodos enriquecidos y número de enlaces asociados a wikidata.

5.2.7 Visualización del grafo enriquecido

El último componente de la solución es la visualización del grafo, para este apartado se usa de neo4j la cual es base de datos orientadas a grafos que permite la serialización de grafos. Neo4j dispone de un complemento que permite el uso de RDF, en el siguiente apartado se muestran las configuraciones necesarias para cargar un grafo RDF en Neo4j.

La primera línea de comando hace referencia a las restricciones y configuraciones del grafo de conocimiento.

```
1. CREATE CONSTRAINT n10s_unique_uri ON (r:Resource) ASSERT r.uri IS UNIQUE;
2. CALL n10s.graphconfig.init({handleVocabUris: 'IGNORE'});
```

La segunda línea de comando importa el archivo. ttl desde el repositorio público frl ptoyecto, en el modelo se utiliza el raw de github para obtener el archivo.

```
3. CALLn10s.rdf.import.fetch("https://raw.githubusercontent.com/accardenas4/Grap
   hEnrich/main/archivo.ttl","Turtle");
```

5.3 Pruebas y resultados

Para ejecutar las pruebas de funcionamiento del modelo planteado y de cada uno de los módulos propuestos en el apartado **4.2 Desarrollo de funcionalidades** se diseñan 3 escenarios para validar el modelo de enriquecimiento. Los casos de uso se desarrollaron en base a temáticas actuales, construyendo triples de información para ser enriquecidos.

A continuación, se detalla cada uno de los casos de uso que serán utilizados para la fase de análisis de resultados. Por cada caso de uso, se realizan dos ejecuciones, la primea ejecución bajo el idioma inglés y la segunda con el idioma español, con el fin de evaluar las herramientas y endpoints utilizados al momento de consultar información.

En la Tabla 7 se resumen los resultados de los tres casos de prueba con los parámetros iniciales que serán utilizados en cada escenario.

Se parte de información que se encuentra alojada en la web bajo la serialización n-triples.

Tabla 7

Casos de uso

#	Dominio	Nodos iniciales	Profundidad	Enlaces de Wikipedia		Nodos enriquecidos	
				Español	Ingles	Español	Ingles
1	**Universidades Ecuador**	65	8	49	56	226	286
2	**Covid19**	7	8	4	2	43	25
3	**Personajes**	90	8	67	67	351	351

5.3.1 Análisis Caso 1

Para el caso de prueba 1 se trabaja sobre el dominio de las universidades del Ecuador, el archivo RDF contiene 65 nodos iniciales, detallados en siguiente enlace.

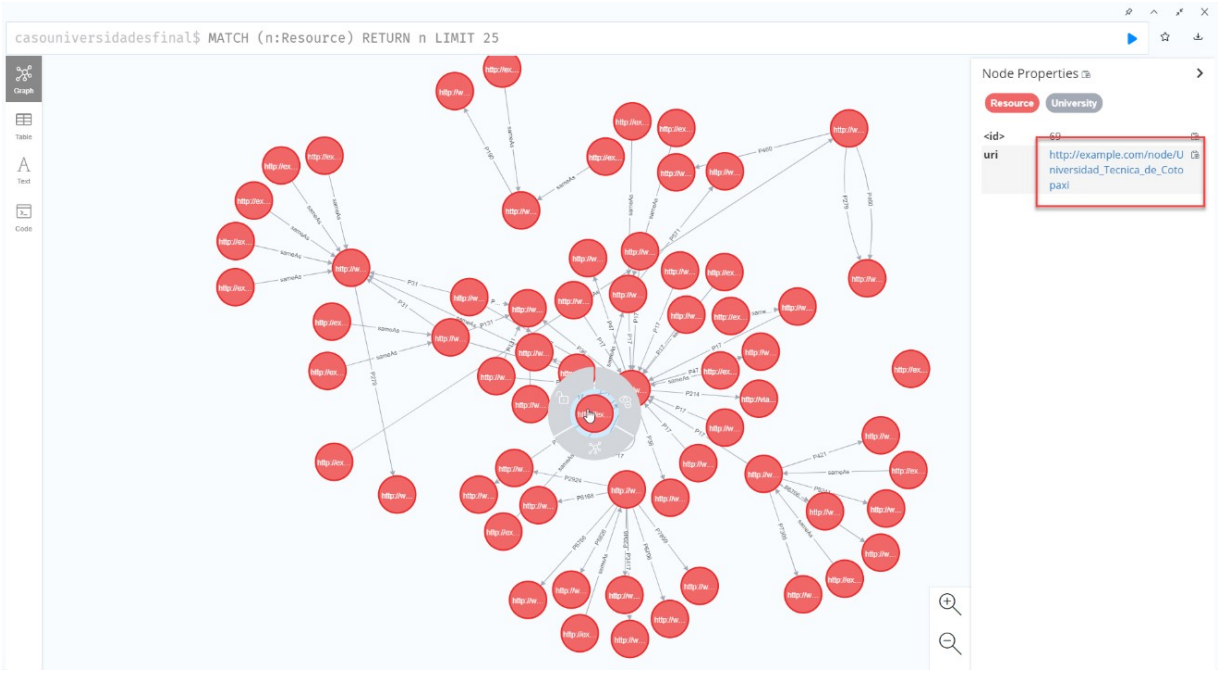
El primer escenario ejecutado se realiza bajo el idioma de inglés y se obtiene 56 referencias a wikipedia y 286 tripletas obtenidas desde wikidata.

En el segundo escenario con el parámetro de idioma en español se observa que se disminuye el número de enlaces encontrados de Wikipedia a 49 y los nodos enriquecidos desde wikidata a 226.

En el análisis realizado sobre el grafo, se evidencia que los nuevos nodos hacen referencia al país, provincia, página web, coordenadas etc. En la Figura 9 se muestra el grafo enriquecido de forma general.

Figura 9

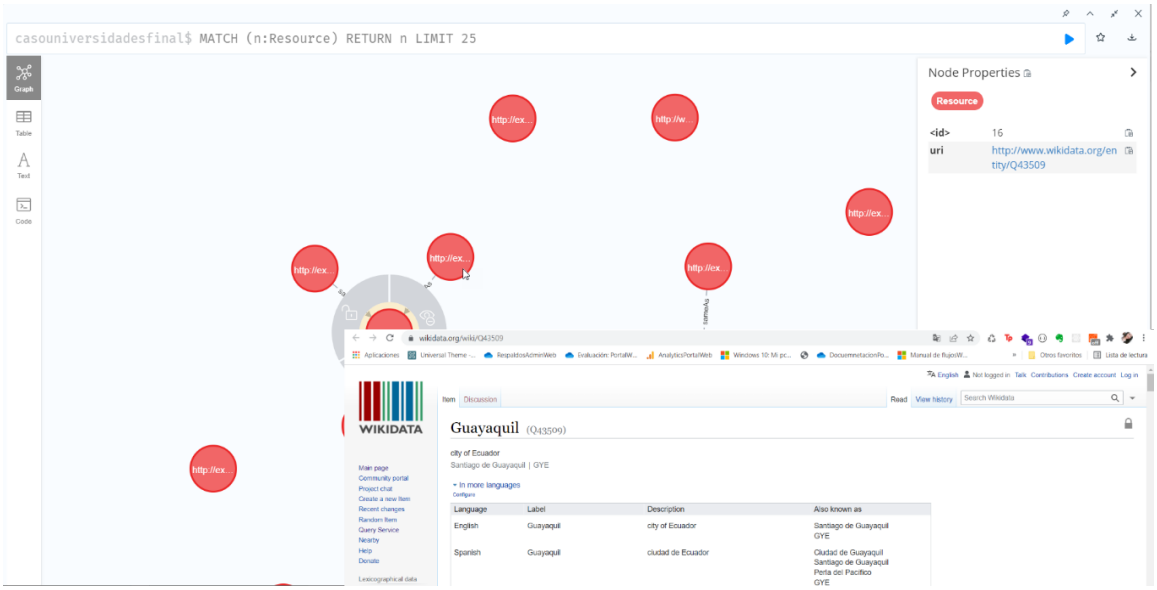
Grafo enriquecido caso 1



Cada uno de los nodos representados en la figura anterior, hacen referencia a enlaces de wikidata, en la Figura 10 se muestra que el recurso Q43509 de Wikipedia que se referencia a la entidad Guayaquil que es la provincia de la universidad católica.

Figura 10

Grafo universidades



En la Tabla 8 se enlistas algunos de los nodos enriquecidos con Wikipedia y la descripción de cada uno de los nodos enriquecidos,

Tabla 8

Caso 1 Nodos enriquecidos

Nodo inicial	Nodos enriquecidos	Descripción
Universidad_de_los_Hemisferios	entity/Q6156836	Universidad asociada a wikidata
	/statement/Q61568360b224f88-4bcb-d863-c610cd4126701caf	Sitio web oficial
	statement/Q6156836-fc06dc244beb-8f5c-6861-227cc68af1f9	Fundación
	statement/Q6156836-c685b61f-419c-8e0b-dabd-a18a8c48cec7	Coordenadas
	entity/Q736	País
	entity/Q2900	Capital
Universidad_Metropolitana_del_Ecuador	/entity/Q30294110	Universidad asociada a wikidata
	entity/Q2385804	Tipo de institución
	entity/Q736	País
Universidad_Particular_San_Gregorio_de_Portoviejo_USGP	entity/Q990716	Provincia

5.3.2 Análisis Caso 2

Para el caso de prueba 2 se trabaja sobre el dominio del virus covid19 sobre un archivo inicial RDF con 7 nodos iniciales, detallados en siguiente enlace. El primer escenario abordado se realiza bajo el idioma de inglés y se obtiene 2 referencias a wikidata y 25 tripletas obtenidas desde wikidata.

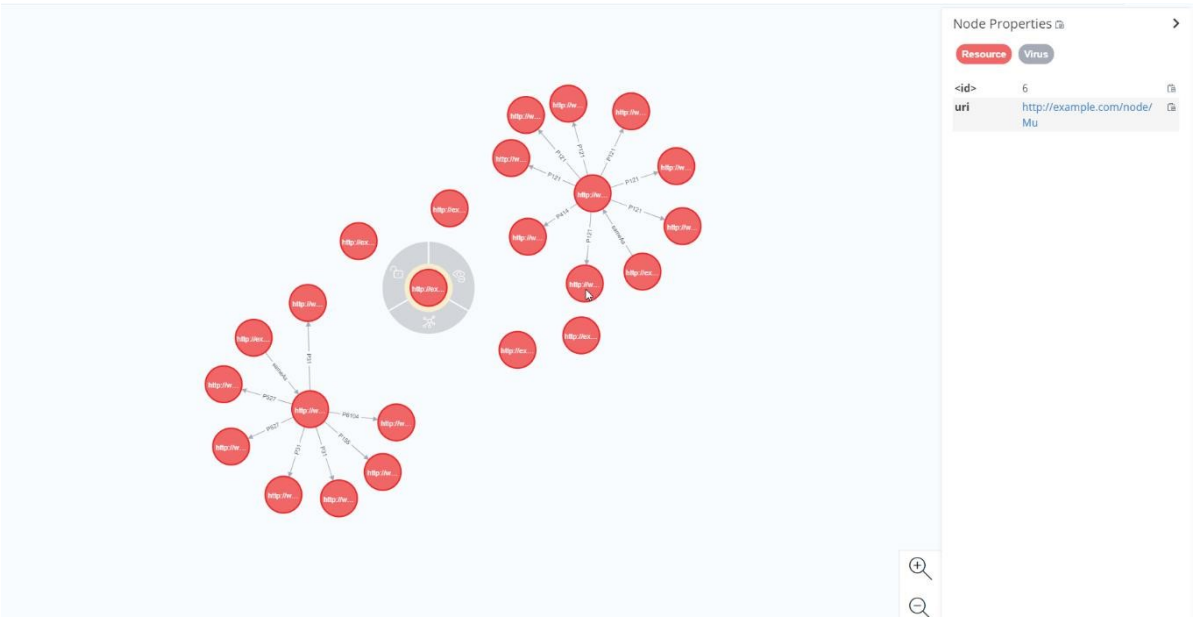
En el segundo escenario con idioma español se duplica el número de recursos encontrados en Wikipedia y 43 tripletas desde wikidata.

En el análisis realizado sobre el grafo, se evidencia que algunos de los nuevos nodos no corresponden con el nodo principal, un caso muy puntual es la variante delta, que dentro del enriquecimiento automático se agrega un nodo referente a una aerolínea. Es decir, los nombres genéricos no son fiables en esta propuesta.

En la Figura 11 se muestra el grafo enriquecido, el nodo seleccionado hace referencia a un nodo aislado que no tuvo enriquecimiento.

Figura 11

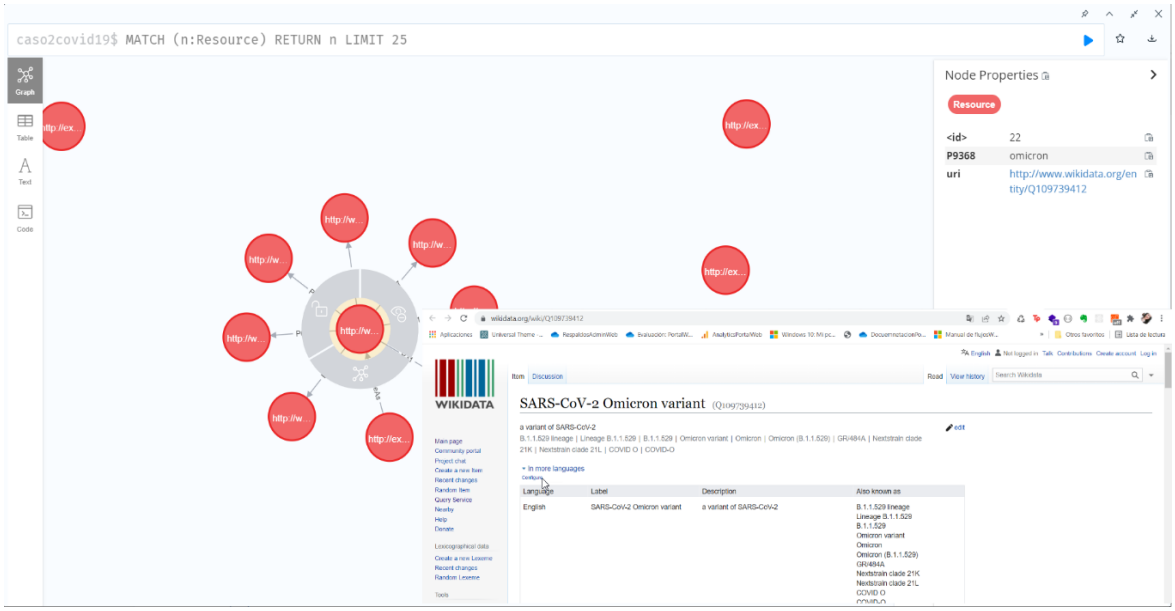
Grafo enriquecido caso 2



Del mismo modo los nodos hacen referencia a enlaces wikidata, en la Figura 12 se muestra la referencia de la variante del coronavirus

Figura 12

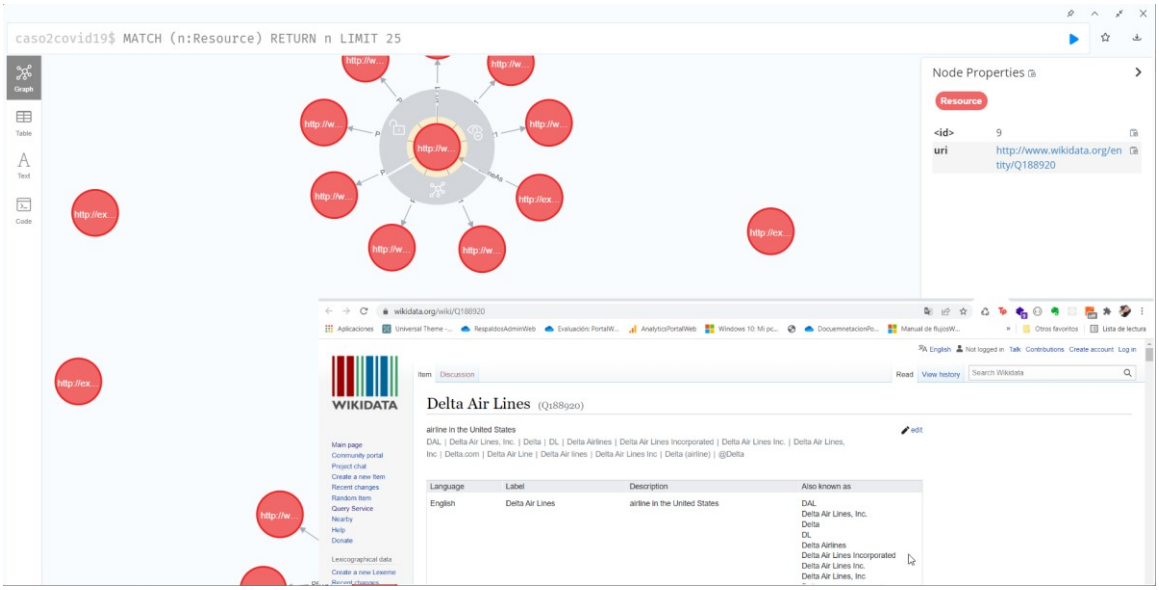
Grafo enriquecido covid19



En este caso se evidencia que existen nodos homónimos que la herramienta de lenguaje natural obtiene de forma errona únicamente cuando se trabaja con el idioma español, en la Figura 13 se muestra que existe el recurso Q188920 en Wikipedia que se denomina Delta Air Lines; sin embargo, este nodo no hace referencia al dominio planteado.

Figura 13

Grafo enriquecido COVID19



En la Tabla 9 se detallan los resultados enriquecidos del grafo inicial, se detalla un nodo, ya que en el apartado anterior se realizó la observación que el gafo presenta homónimos que entorpecen el enriquecimiento del graf, el grafo enriquecido se encuentra en el siguiente enlace

Tabla 9

Resultados caso 2

Nodo inicial	Nodos enriquecidos	Descripción
Omicron	entity/Q109739412	Variante asociada a wikidata
	entity/Q107143124	Variantes activas
	entity/Q105758262	Variantes con alto riesgo
	entity/Q104450895	Variante asociada

5.3.3 Análisis Caso 3

Para el caso de prueba número 3 se trabaja sobre el dominio de personas en base un archivo inicial RDF que cuenta con 90 nodos iniciales, detallados en el Anexo 3,

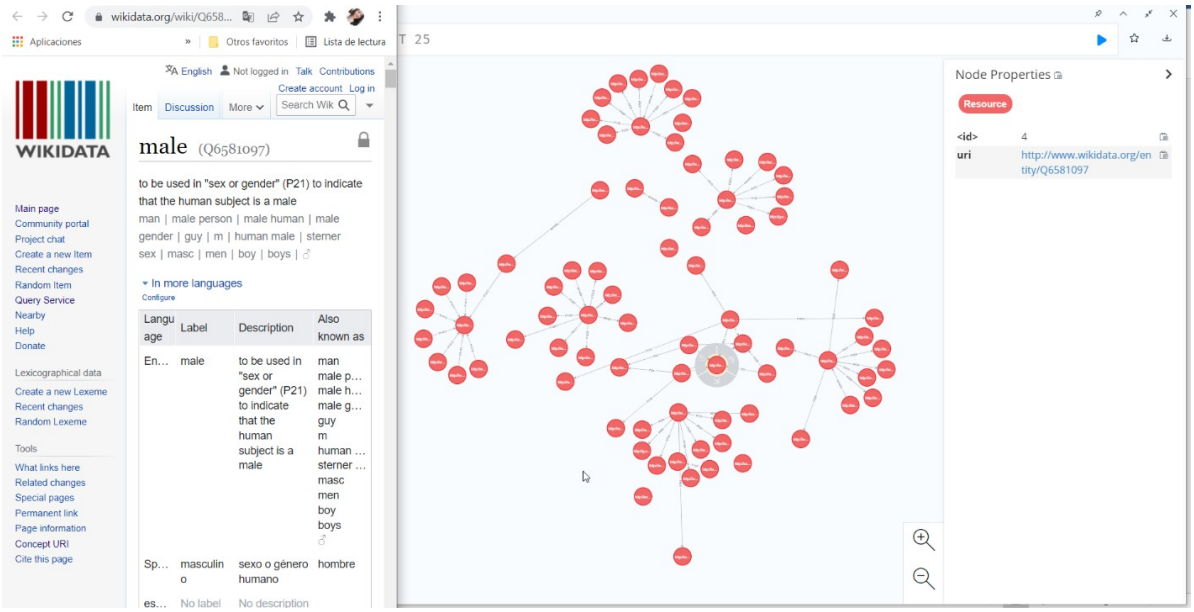
En el análisis realizado sobre el grafo, se evidencia que todos los nuevos nodos plegados corresponden al nodo origen, en el primer escenario abordado se realiza la prueba con el idioma ingles obteniendo un resultado favorable de 67 nodos encontrados en Wikipedia y 351 nuevas tripletas desde wikidata.

En el segundo escenario se aborda el lenguaje español donde se obtiene los mismos resultados que el primer escenario, bajo este dominio el comportamiento es similar.

En el análisis realizado del grafo, se evidencia que los nuevos nodos hacen referencia al país, idioma, género, tipo de política etc. En la Figura 15 se muestra el grafo enriquecido en neo4j.

Figura 14

Grafo enriquecido caso 3



Algunos de los nodos enriquecidos por persona se encuentran el enlace de wikidata asociado, el cargo, honorarios, fecha de nacimiento, defunción, etc.

5.4 Discusión de resultados

En base al análisis realizado por cada escenario, el modelo cumple parcialmente con el enriquecimiento de los datos:

- Los análisis de entidades en algunos escenarios no se encuentran las relaciones asociadas a wikidata.
- Se dificulta proceso de análisis de entidades genéricas dentro del modelo, en el caso dos existe un nodo que no corresponde al grafo inicial, dado que existen homónimos de la entidad a enriquecer.

Por otra parte, se evidencia que el resultado obtenido del análisis de entidades a través de procesamiento de lenguaje natural correspondiente al escenario 1 se obtiene un 86% de reconocimiento de entidades en Wikipedia, correspondiente al grafo inicial Se muestra que el uso del idioma impacta (inglés o español) en el enriquecimiento del grafo y en las consultas sobre el endpoint SPARQL. En idioma ingles se encuentran más resultados para el enriquecimiento.

Hacer uso de parámetros como la lista negra de predicados permite realizar un enriquecimiento limpio de los datos que se necesitan obtener desde Wikipedia.

Conclusiones

Después de finalizar el trabajo de fin de master y cumpliendo los objetivos planteados, se han tenido los siguientes resultados:

Investigación de trabajos relacionados, el cual permitió conocer indicadores faltantes dentro de las investigaciones de los 5 últimos años, permitiendo plantear una solución viable para el enriquecimiento de grafos.

Desarrollo de un modelo capaz de cumplir con los objetivos planteados, para el desarrollo se usan herramientas que facilitan el proceso de enriquecimiento de grafos

Desarrollo de un prototipo capaz de generar nodos candidatos desde fuentes de datos abiertas, el prototipo se enfocó a wikidata.

Planteamiento de casos de uso para medir el rendimiento de la solución planteada

Una vez culminado el trabajo se concluye lo siguiente:

Para el enriquecimiento de un grafo de conocimiento se debe considerar procesos automáticos que faciliten la clasificación de entidades y procesos semiautomáticos con el fin de tomar decisiones en casos de encontrar múltiples coincidencias sobre un nodo.

El uso de herramientas que manejen procesamiento de lenguaje natural facilitan el enriquecimiento de grafos de conocimiento, es decir, hacer uso de redes entrenadas disminuyen el desarrollo facilitando para analizar entidades como primer paso para enriquecer un grafo.

La mayoría de los grafos de conocimiento que se encuentran en la web son grafos etiquetados. Sin embargo, los grafos RDF cumplen con el principio de utilizar un lenguaje común para hacer referencia a los datos.

Recomendaciones

A continuación, se presentan las recomendaciones generadas en base a los resultados obtenidos:

Bajo el modelo planteado, se debe continuar investigando y desarrollando funcionalidades que permita alertar al usuario final si el nodo enriquecido realmente tiene una relación con la entidad origen.

Desarrollar un mecanismo capaz de solucionar el problema actual de homónimos dentro de las entidades de cada grafo.

En base al prototipo planteado se debe mejorar el algoritmo con el fin de permitir asociar datos estructurados y no solo datos abiertos como Wikipedia, utilizando procesamiento de lenguaje natural

Desarrollar y mejorar funcionalidades que permitan el enriquecimiento a grafos de propiedades, permitiendo abordar a las dos representaciones de conocimiento.

La propuesta de enriquecimiento planteada hace uso de la nube de google; sin embargo, se recomienda realizar una comparación de lenguaje natural con diferentes herramientas, con el fin de medir el rendimiento.

Referencias

Abu-Salih, B., Al-Tawil, M., Aljarah, I., Faris, H., Wongthongtham, P., Chan, K. Y., & Beheshti, A. (2021). Relational Learning Analysis of Social Politics using Knowledge Graph Embedding. In *Data Mining and Knowledge Discovery* (Vol. 35). https://doi.org/10.1007/s10618-021-00760-w

Andreotti, R., Emmanuele, A., Fontanella, D., Zanier, F., & Luise, M. (2015). Translating Embeddings for Modeling Multi-relational Data. *2014 7th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing, NAVITEC 2014 - Proceedings*, 1–9. https://doi.org/10.1109/NAVITEC.2014.7045139

Angles, R. (2018). The property graph database model. *CEUR Workshop Proceedings*, *2100*(Section 2).

Arenas, M., Cuenca Grau, B., Kharlamov, E., Marciuška, Š., & Zheleznyakov, D. (2016). Faceted search over RDF-based knowledge graphs. *Journal of Web Semantics*, *37–38*, 55–74. https://doi.org/10.1016/j.websem.2015.12.002

Arnaout, H., & Elbassuoni, S. (2018). Effective searching of RDF knowledge graphs. *Journal of Web Semantics*, *48*(2017), 66–84. https://doi.org/10.1016/j.websem.2017.12.001

Bartscherer, F., Menne, C., & Rettinger, A. (2017). *KG-Comparison-SWJ-Article*. *0*.

Cao, E., Wang, D., Huang, J., & Hu, W. (2020). Open Knowledge Enrichment for Long-tail Entities. *The Web Conference 2020 - Proceedings of the World Wide Web Conference, WWW 2020*, *2*, 384–394. https://doi.org/10.1145/3366423.3380123

Dettmers, T., Minervini, P., Stenetorp, P., & Riedel, S. (2018). Convolutional 2D knowledge graph embeddings. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 1811–1818.

Galkin, M. (2020). Representation Learning on RDF* and LPG Knowledge Graphs. Retrieved from https://towardsdatascience.com/representation-learning-on-rdf-and-lpgknowledge-graphs-6a92f2660241

Gharibi, M., Zachariah, A., & Rao, P. (2020). FoodKG: A Tool to Enrich Knowledge Graphs Using Machine Learning Techniques. *Frontiers in Big Data*, *3*(June), 3389. https://doi.org/10.3389/fdata.2020.00021

Heiko Paulheim. (2016). Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods. *Semantic Web*, *8*(3), 489–508.

Manzano, I. (2015). Knowledge Graph: una nueva fuente de conocimiento. Retrieved from https://medium.com/@ira.manzano/knowledge-graph-una-nueva-fuente-deconocimiento-ca645a77fd68

Martínez Arellano, F. F., & Amaya Ramírez, M. Á. (2017). El papel de los metadatos en la Web Semántica. *Biblioteca Universitaria*, *20*(1), 3–10. https://doi.org/10.22201/dgb.0187750xp.2017.1.171

Mauthner, N. S., & Parry, O. (2013). Open Access Digital Data Sharing: Principles, Policies and Practices. *Social Epistemology*, *27*(1), 47–67. https://doi.org/10.1080/02691728.2012.760663

Mittal, N. M., & Choudhary, S. (2014). *Comparative Study of Simulators for Vehicular Ad-hoc Networks (VANETs)*. *4*(4).

Muhan Zhang. (2018). Link Prediction Based on Graph Neural Networks. *Notes and Queries*. https://doi.org/10.1093/nq/s7-I.23.447-i

Murray-Rust, P. (2008). Open data in science. *Serials Review*, *34*(1), 52–64. https://doi.org/10.1080/00987913.2008.10765152

Petzold, R., Gesese, G. A., Bogdanova, V., Zylowski, T., Sack, H., & Alam, M. (n.d.). *Challenges of Applying Knowledge Graph and their Embeddings to a Real-world Usecase*.

Rossi, A., Barbosa, D., Firmani, D., Matinata, A., & Merialdo, P. (2021). Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data*, *15*(2). https://doi.org/10.1145/3424672

Saeed, M. R., Chelmis, C., & Prasanna, V. K. (2019). Extracting entity-specific substructures for RDF graph embeddings. *Semantic Web*, *10*(6), 1087–1108.

https://doi.org/10.3233/SW-190359

Saorín, T. (2019). Grafos de conocimiento y bases de datos en grafo: conceptos fundamentales a partir de una “obra maestra” del Museo del Prado. *Anuario ThinkEPI*, *13*, 1–7. https://doi.org/10.3145/thinkepi.2019.e13f05

Tello, A. (2020). Ontologías en la web semántica. *Departamento de Informática de La Universidad De*, 1–4. Retrieved from http://www.anobium.es/docs/gc_fichas/doc/68ERfhjkmv.pdf

Tempelmeier, N., & Demidova, E. (2021). Linking OpenStreetMap with knowledge graphs — Link discovery for schema-agnostic volunteered geographic information. *Future Generation Computer Systems*, *116*, 349–364. https://doi.org/10.1016/j.future.2020.11.003

Torres-Carrion, P. V., Gonzalez-Gonzalez, C. S., Aciar, S., & Rodriguez-Morales, G. (2018). Methodology for systematic literature review applied to engineering and education. *IEEE Global Engineering Education Conference, EDUCON*, *2018-April*(April), 1364–1373. https://doi.org/10.1109/EDUCON.2018.8363388

Trouillon, T., Welbl, J., Riedel, S., Ciaussier, E., & Bouchard, G. (2016). Complex embeddings for simple link prediction. *33rd International Conference on Machine Learning, ICML 2016*, *5*, 3021–3032.

Yang, B., Yih, W. tau, He, X., Gao, J., & Deng, L. (2015). Embedding entities and relations for learning and inference in knowledge bases. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–12.

Zhao, Z., Han, S.-K., & So, I.-M. (2018). Architecture of Knowledge Graph Construction Techniques. *International Journal of Pure and Applied Mathematics*, *118*(19), 1869–1883.