UNIVERSIDAD NACIONAL DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

Programa de Apoyo al Egreso para Alumnos con Práctica Profesional Supervisada

TÍTULO: Detección de duplicados sobre grafos de conocimiento de avisos inmobiliarios

AUTOR/A: Felipe Dioguardi

DIRECTOR/A ACADÉMICO: Dr. Diego Torres

DIRECTOR/A PROFESIONAL: Dr. Juan Pablo del Río

CODIRECTOR/A ACADÉMICO: Dr. Leandro Antonelli

CARRERA: Licenciatura en Informática

RESUMEN

Este trabajo es una contribución al proyecto "Observatorio de valores del suelo e instrumentos de financiamiento del desarrollo urbano", cuyo objetivo es tener una herramienta que sirva para analizar el mercado inmobiliario en la provincia de Buenos Aires y así desarrollar políticas públicas de urbanización más efectivas. Esta investigación se enfoca en identificar entidades duplicadas en un grafo de conocimiento sobre el sector inmobiliario que servirá como base de datos principal de la herramienta. El grafo está estructurado mediante una ontología del dominio inmobiliario, y poblado mediante el uso de web scraping sobre distintas páginas de avisos inmobiliarios. La construcción del grafo se describe en este trabajo para contextualizar sobre el origen de la existencia de los duplicados. Dado que la existencia de entidades duplicadas puede obstaculizar el análisis estadístico de la información debido a las inconsistencias que generan, este trabajo propone una solución basada en un enfoque bayesiano y evaluada sobre un conjunto de los datos reales curado a mano por los expertos en el dominio.

Palabras Claves

Detección de duplicados, grafo de conocimiento, avisos inmobiliarios, clasificador bayesiano, web scraping.

Conclusiones

Al finalizar esta investigación, se logró consolidar un grafo de conocimiento con información actualizada sobre el mercado inmobiliario y una ontología capaz de describir el dominio en cuestión. Se detectó el 70.4% de las entidades duplicadas en el grafo mediante una técnica de detección de duplicados con un enfoque bayesiano con una precisión del 66.8%, evaluada con un conjunto de datos reales etiquetados por expertos en el dominio.

Trabajos Realizados

En este trabajo se consolidó un grafo de conocimiento que describe el mercado inmobiliario de la provincia de Buenos Aires a partir de datos provenientes de diversas páginas web. Se diseñó una ontología para modelar esta información teniendo en cuenta los vocabularios existentes en la Web. Se relevó el estado del arte en técnicas de detección de duplicados, y se utilizó un Clasificador Bayesiano para uniformar las múltiples referencias a una misma entidad.

Trabajos Futuros

Se proponen como trabajos a realizar a partir de esta tesina:

- *La exploración de configuraciones alternativas sobre la técnica empleada.*
- *La implementación de herramientas de detección de duplicados basadas en otras técnicas, así como graph embedding, inteligencia artificial, y el análisis estructural de grafos.*

Fecha de la presentación: MES Y AÑO

Detección de duplicados sobre grafos de conocimiento de avisos inmobiliarios

Felipe Dioguardi

LIFIA, CICPBA-Facultad de Informática, UNLP
felipe.dioguardi@lifia.info.unlp.edu.ar

Resumen El problema de la duplicación en grafos de conocimiento surge cuando existen múltiples nodos que no están vinculados entre sí, pero describen la misma entidad. La duplicación en las bases de conocimiento es un problema porque implica la ambigüedad semántica de los datos, llevando a contradicciones, inconsistencias, y errores en los análisis. Este trabajo releva el estado del arte de las tecnologías, algoritmos y técnicas de detección de duplicados aplicables sobre grafos de conocimiento, y presenta una estrategia aplicada a un grafo de conocimiento que contiene información inmobiliaria extraída de diferentes sitios web. El problema de la detección de duplicados se trata como un caso de *instance matching* aplicado a un único grafo de conocimiento. Además, se utiliza un clasificador bayesiano que compara similitudes sintácticas entre registros para identificar relaciones `owl:sameAs` implícitas. Se propone segmentar la comparación según los atributos implicados y asignar diferentes pesos a cada segmento. La eficacia de la estrategia se evalúa en el contexto del dominio inmobiliario, donde este tipo de duplicaciones aparecen en gran medida. Para esto, se utiliza un subconjunto de datos del mundo real seleccionado por expertos, con la forma de un grafo de conocimiento de avisos inmobiliarios con entidades duplicadas y únicas. Este enfoque consiguió limpiar el grafo de conocimiento con una precisión del 66,8 %, una exhaustividad del 70,4 % y un Valor-F del 68,6 %.

Palabras clave: Detección de duplicados · Grafo de conocimiento · Avisos inmobiliarios · Clasificador bayesiano · Web scraping

1. Introducción

El mercado inmobiliario desempeña un papel fundamental en el crecimiento económico y social de cualquier nación. Sin embargo, el acceso a la información sobre sus características, precios y tendencias es limitado y disperso. Esto supone un desafío tanto para los agentes públicos como para los privados a la hora de tomar decisiones informadas. Por ello, la Comisión de Investigaciones Científicas (CIC) y el Organismo Provincial de Integración Social y Urbana (OPISU), se propusieron crear un Observatorio de Valores de Suelo que le permita monitorear y analizar el comportamiento del sector inmobiliario en la provincia de Buenos Aires. [1].

[1] Observatorio de Valores del Suelo (OVS): https://observatoriosuelo.gba.gob.ar/

Para lograr este objetivo se necesitan datos actualizados y fiables sobre la oferta y la demanda de bienes inmuebles en la provincia. Sin embargo, obtener cantidades sustanciales de esta información resulta complicado, ya que a menudo no está estructurada, es inaccesible o no está disponible. Una de las fuentes de información más significativas y utilizadas en el mercado inmobiliario son las páginas web de anuncios de oferta inmobiliaria, que brindan una gran cantidad de datos sobre los inmuebles ofertados, tales como su ubicación, superficie, precio, tipo, estado, descripción, fotos, etc.

Para extraer estos datos de los sitios web, una técnica aplicable es el *web scraping*, que consiste en la extracción automática de conocimiento de páginas web mediante software. El *web scraping* permite obtener grandes volúmenes de datos de forma rápida y eficaz, pero existen varios retos y limitaciones que deben tenerse en cuenta y abordarse para garantizar la calidad y utilidad de los datos obtenidos.

Uno de los principales problemas que se encuentran al hacer *scraping* de sitios web de anuncios inmobiliarios es la existencia de duplicados, es decir, registros que corresponden al mismo inmueble, pero proceden de diferentes sitios web o de diferentes publicaciones dentro de la misma plataforma. En el mercado inmobiliario, un mismo inmueble puede aparecer en varios anuncios. Esto puede ocurrir si, por ejemplo, más de una agencia inmobiliaria ofrece el mismo inmueble, y cada una de las agencias lo publica en distintos anuncios del mismo sitio. Esto mismo puede ocurrir si las agencias publican el mismo inmueble en sitios diferentes. Además, anuncios completamente diferentes pueden tener descripciones similares debido a las plantillas utilizadas por las agencias inmobiliarias u ofrecidas por las mismas plataformas. Las coordenadas geográficas de los inmuebles ofertados también pueden suponer un reto, dependiendo de quién y cómo los publique. Asimismo, las convenciones de nomenclatura pueden variar según las fuentes, lo que dificulta la identificación de duplicados sin un enfoque exhaustivo que tenga en cuenta todos los atributos relevantes de cada anuncio.

La presencia de duplicados puede introducir incoherencias, redundancias e imprecisiones que pueden dar lugar a errores o sesgos en los análisis posteriores [3]. Eliminar los registros duplicados de una base de conocimiento es fundamental para garantizar datos fidedignos [22], ya que su presencia puede perjudicar el análisis estadístico y socavar sus resultados.

Para evitar estos problemas, es necesario aplicar técnicas de deduplicación, que consisten en identificar y eliminar los registros duplicados de una base de conocimiento. La detección de duplicados, a menudo *record linkage*, *entity resolution*, y deduplicación de datos, es el proceso de identificación de diferentes registros que hacen referencia a la misma entidad [8]. Este problema no es trivial, ya que implica algunos pasos desafiantes, desde la cuidadosa selección de los pares de registros a comparar hasta el método de comparación de estos registros y el proceso final de vinculación. La deduplicación es un proceso complejo y difícil que requiere métodos y algoritmos adecuados para comparar y unir registros de forma eficiente y precisa. La deduplicación también implica tomar decisiones

sobre qué criterios utilizar para determinar si dos registros son duplicados, y qué información conservar o descartar de cada registro en caso afirmativo.

Se han propuesto numerosas herramientas y estrategias para abordar la detección de duplicados. Entre las estrategias para identificar posibles coincidencias se encuentran las técnicas de vectorización [26], que resultan muy prometedoras y se estudiarán en el futuro, y las consultas con *fuzzy matching* [14], utilizadas en este trabajo. Asimismo, se han explorado modelos de aprendizaje automático [2] y sistemas basados en reglas [23] para comparar registros potencialmente duplicados. Los primeros aspiran a convertirse en un estándar por sus capacidades y avances recientes, pero requieren un entrenamiento exhaustivo para que los sistemas sean competentes. Además, suelen funcionar como una caja negra, lo que limita la transparencia y la facilidad de modificación. Por otro lado, los sistemas basados en reglas requieren una considerable configuración por parte de expertos y una gran cantidad de datos de entrada iniciales para funcionar como se desea. La generación y el mantenimiento de reglas exigen un esfuerzo notable, lo que dificulta su aplicación práctica. Por otra parte, se han desarrollado frameworks que automatizan todos los aspectos de la deduplicación de entidades [35]. Sin embargo, el éxito de estos métodos depende en gran medida de su adaptabilidad a las características específicas de cada dominio, y la selección de un método adecuado puede verse influida por factores como el tamaño y la calidad de los datos, la naturaleza de las funciones de comparación y los requisitos de la aplicación.

Esta investigación utiliza una estrategia de detección de duplicados que aprovecha el Teorema de Bayes para agregar probabilidades, aplicada a una base de conocimiento de anuncios inmobiliarios. Para ello, evalúa el rendimiento de una estrategia de detección de duplicados basada en un clasificador bayesiano sobre una base de conocimiento de ofertas inmobiliarias construida a partir de información extraída de diferentes sitios web. El rendimiento de este enfoque se evalúa utilizando las métricas de precisión, exhaustividad y Valor-F, y se asegura una prueba insesgada mediante el uso de una base de conocimiento y una verdad fundamental (*ground truth dataset*) con datos curados manualmente.

Este trabajo se organiza de la siguiente forma. La Sección 2 presenta la motivación para realizar este trabajo en el marco del proyecto del OVS. La Sección 3 expone el estado del arte y el marco de las investigaciones previas sobre la detección de duplicados. En la Sección 4 se define el problema de la detección de duplicados y se analizan los retos particulares relacionados con el dominio de los avisos inmobiliarios. En la Sección 5 se detalla el grafo de conocimiento, incluidas las fuentes y las técnicas de extracción y modelado del conocimiento empleadas en el estudio. En la Sección 6 se presenta en detalle el enfoque bayesiano de detección de duplicados utilizado en este estudio, y en la Sección 7 se describen las métricas aplicadas para evaluar su eficacia. A continuación, la Sección 8 presenta la metodología empleada para la evaluación de esta técnica, mientras que la Sección 9 presenta los resultados y un análisis de los mismos. Por último, la Sección 10 presenta las conclusiones de la evaluación y

ofrece una visión de los puntos fuertes y débiles relativos de los distintos enfoques, al tiempo que analiza posibles áreas de investigación futura.

2. Motivación

En agosto de 2021, el Ministerio de Ciencia, Tecnología e Innovación de la Nación aprobó el proyecto denominado "Observatorio de valores del suelo e instrumentos de financiamiento del desarrollo urbano". Este tiene como objetivo principal resolver la carencia estructural en la disponibilidad pública de información estratégica de valores de mercado inmobiliario para cuantificar las valorizaciones producidas por la acción del Estado, en el marco de la política de Integración social y urbana de barrios populares [28].

Para llevarlo a cabo, el Organismo Provincial de Integración Social y Urbana (OPISU) propuso la creación de un observatorio de valores del suelo, una herramienta que permita la producción y sistematización de datos provenientes del mercado inmobiliario. A partir del estudio de esa información y del desarrollo de instrumentos de recuperación de plusvalías urbanas, se espera contribuir al sostenimiento a mediano y largo plazo del hábitat de los barrios populares, y fortalecer los procesos de integración socio-urbana. La creación del OVS fue delegada al Laboratorio de Investigación del Territorio y el Ambiente de la Comisión de Investigaciones Científicas (LINTA-CIC), formado por un grupo de urbanistas, geógrafos, arquitectos, y expertos en estadística encargados de diseñar las estrategias de análisis de la información inmobiliaria.

En las instancias iniciales del proyecto, los investigadores realizaron un balance de las diversas fuentes de información disponibles, clasificándolas según la cantidad y calidad de los avisos publicados sobre algunos partidos de interés de la provincia de Buenos Aires. A partir de esa evaluación, seleccionaron diferentes sitios de ofertas inmobiliarias, y recabaron manualmente información acerca de cientos de inmuebles publicitados en ellos. Para poder efectuar un estudio completo del valor del suelo en las distintas áreas de la provincia, era necesario tener la capacidad de extraer grandes volúmenes de datos de manera sistemática. Resultó natural pensar en la construcción de un programa informático que automatizara este proceso, con el fin de aliviar el trabajo de los investigadores y permitirles centrarse en las etapas siguientes.

Web scraping es una técnica para obtener información de páginas de Internet, y almacenarla en un archivo o base de datos local para su posterior análisis [34]. A su vez, un *web crawler* o *spider* es un agente informático utilizado para la descarga masiva de páginas web [29,31]. Ambos conceptos suelen acompañarse, pues es común querer recolectar en una misma estructura de datos el conocimiento contenido en un gran conjunto de páginas web.

Una herramienta de *web scraping* con la capacidad de acceder a todos los avisos inmobiliarios útiles de los sitios seleccionados fue considerada la alternativa ideal para resolver la tarea propuesta. La herramienta debía, además, normalizar y estructurar los datos obtenidos, para garantizar que el análisis consecuente

pueda llevarse a cabo. Esta necesidad surgió del carácter heterogéneo inherente a las diversas páginas de la Web.

Un modo de formalizar el conocimiento recuperado es a través de ontologías [4]. Se define como ontología a una descripción del conocimiento sobre un dominio de interés, cuyo núcleo es una especificación procesable por las máquinas con un significado formalmente definido [19]. Definir una ontología para avisos inmobiliarios no solo permitiría establecer un esquema riguroso para representar la información, sino que también la dotaría de valor semántico que podrá ser aprovechado para su curado.

Una problemática frecuente a tener en cuenta es que un mismo inmueble suele estar publicado múltiples veces, tanto a través de distintas plataformas, como bajo el cargo de diferentes empresas inmobiliarias. Para sacar el máximo provecho a la información recolectada, era necesario también unificar el conocimiento obtenido sobre un mismo inmueble. Los datos nutridos por múltiples ofertas aportarían sin duda una mayor precisión al cálculo del valor del suelo en el estudio posterior.

La detección de duplicados en bases de conocimiento es un problema común en una variedad de dominios. En la revisión literaria de Huaman et al. [21] se describen numerosas técnicas y herramientas creadas para solucionarlo, particularmente sobre bases de conocimiento estructuradas en forma de grafo. Un grafo de conocimiento es una estructura de datos diseñada para almacenar conocimiento sobre determinados dominios. Esta información puede ser heterogénea y provenir de fuentes distintas. En los grafos de conocimiento, los nodos representan las entidades de interés, y las aristas representan las relaciones entre esas entidades [11,20]. Por este motivo, se planteó que la herramienta propuesta almacene la información en un grafo de conocimiento, sobre el cual poder aplicar una técnica de deduplicación que aumenten la calidad de los datos.

Para eso fue necesario realizar un relevamiento del estado del arte de las estrategias de detección de duplicados disponibles, con el fin de determinar cuál se adapta mejor al inconveniente que se presenta en este dominio en particular.

3. Trabajos Relacionados

Esta sección presenta los trabajos relacionados sobre el problema de la detección de duplicados. Varios de ellos han analizado la literatura sobre herramientas de detección de duplicados, como Elmangarmid et al. [8] en el contexto de bases de datos, Assi et al. [1] con su estudio sobre sistemas de resolución de *instance matching*, y Huaman et al. [21] con su revisión del estado del arte para la detección de duplicados en grafos de conocimiento.

La deduplicación probabilística tiene sus raíces en la teoría introducida por Fellegi y Sunter [10], quienes establecen un marco formal para definir las hipótesis de emparejamiento y no emparejamiento, y para calcular las probabilidades posteriores de cada una. Una de las asunciones clave de esta teoría es que los atributos de los registros son independientes entre sí, lo que simplifica el cálculo de las probabilidades, pero también puede introducir errores de emparejamiento.

El enfoque bayesiano ha sido aplicado y extendido por varios autores para resolver diferentes problemas de *entity resolution*. Binette et al. [5] proporcionan una visión histórica de la evolución de los métodos de detección de duplicados, explicando los mecanismos que surgieron para resolverlo, desde los determinísticos como los enfoques basados en reglas y similitud, hasta los modelos probabilísticos más recientes. Dentro de los más explorados en este último grupo se encuentra el enfoque Bayesiano, que, como explica [30], tuvo una evolución por sí mismo dentro de la literatura. Al surgir, se desarrollaban modelos específicos para comparar cada atributo de un registro. Sin embargo, estos modelos solo tenían en cuenta variables nominales o numéricas. Esto no era viable en la práctica porque comúnmente se buscaba distinguir registros de distintos tipos, longitudes, y significados, como nombres, direcciones, fechas y más. Para solucionar este problema surgió la comparación entre pares de registros. Esto dio paso a vincular registros no solo explícitamente mediante comparaciones, sino implícitamente gracias a la propiedad transitiva de la igualdad. Enamorado et al. [9] reportan su experiencia en la deduplicación de una base de datos con millones de registros de votantes, utilizando un modelo bayesiano que compara todos los atributos con la distancia de Jaro-Winkler, y que ignora los atributos faltantes. Sin embargo, este método no considera que la distancia de Jaro-Winkler puede no ser la más adecuada para todos los atributos.

Diversos estudios han examinado diferentes métodos para resolver la detección de duplicados con el uso de un enfoque bayesiano. Liseo y Tancredi [25] evalúan los avances en la metodología bayesiana para la vinculación de registros y la inferencia con las unidades emparejadas, enfatizando el uso de técnicas de simulación y muestreo para obtener estimaciones precisas y eficientes. Liseo y Tancredi [33] también proponen un modelo estadístico jerárquico para el emparejamiento de duplicados, que tiene en cuenta el tamaño de la población y la heterogeneidad de los registros. Dong et al. [7] estudian cómo detectar dependencias entre fuentes de datos mediante modelos bayesianos, que analizan los valores compartidos entre fuentes y que representan los resultados como una base de datos probabilística, en donde cada objeto se asocia a una distribución de probabilidad de varios valores en el dominio subyacente.

4. Definición del Problema

Esta sección ahonda en el problema de la detección de duplicados en grafos de conocimiento, examinando su complejidad y la estructura común de las posibles soluciones. En primer lugar, se introducirán conceptos esenciales relacionados con la representación del conocimiento y la definición de duplicados.

Una base de conocimientos es una colección de conceptos relacionados con un dominio específico que puede almacenarse en diversos formatos [32,20,19], como XML o RDF. Estos formatos permiten crear un grafo de conocimiento (KG por sus siglas en inglés), que representa el conocimiento como una red de entidades unidas por relaciones, capturando sus relaciones semánticas y permitiendo descubrir nuevo conocimiento a través de vínculos implícitos. Como cualquier

estructura de grafos, los grafos de conocimiento están formados por nodos, cada uno de los cuales representa un concepto único. Para mantener esta individualidad, los nodos se identifican mediante Identificadores Internacionales de Recursos (IRI por sus siglas en inglés), que garantizan que cada concepto tenga un nombre único. En la Web existen vocabularios y ontologías, que proveen acceso único a diversas IRIs que representan conceptos reutilizables en varios grafos de conocimiento. Las ontologías desempeñan un papel fundamental al proporcionar un marco formal y estructurado para representar conceptos y sus relaciones dentro de un dominio. Al definir el vocabulario y las reglas para describir entidades y sus interacciones, las ontologías fomentan la interoperabilidad entre diversas bases de conocimiento, facilitando el intercambio fluido entre distintos dominios.

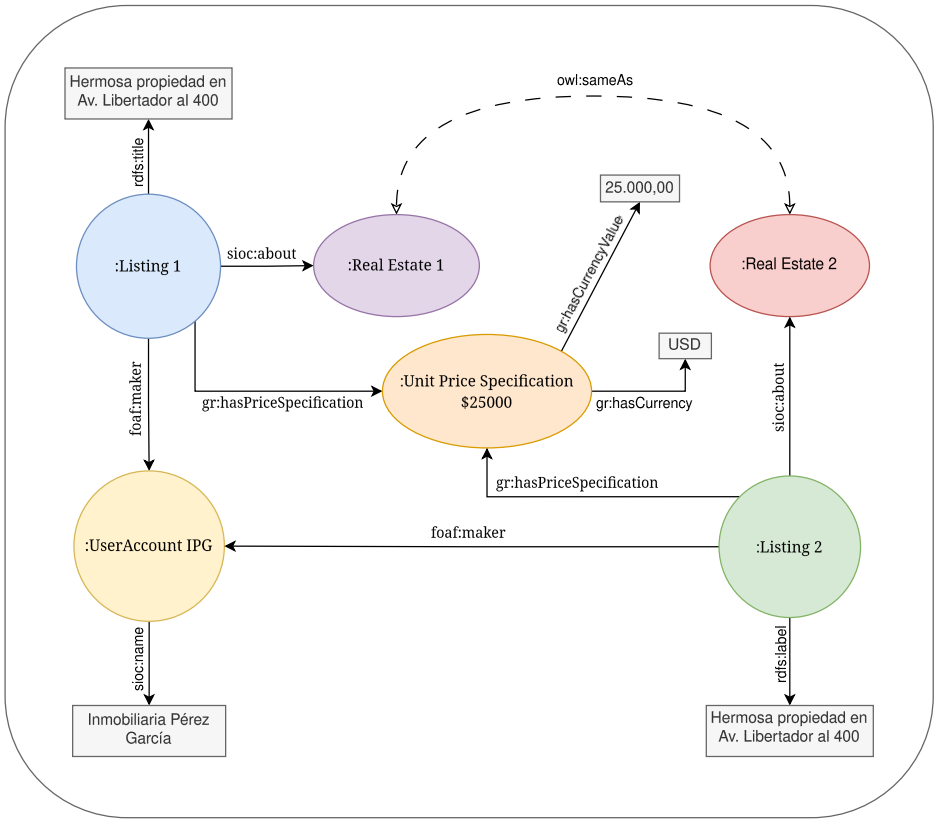

Figura 1. Duplicados exactos en un grafo de conocimiento.

Los duplicados en los grafos de conocimiento se producen cuando dos nodos que representan a la misma entidad tienen identificadores diferentes. Los nodos que comparten las mismas propiedades, pero difieren en sus identificadores, se denominan duplicados exactos. Por otro lado, los nodos que difieren

en algunas o todas sus propiedades, pero que se refieren a la misma entidad, se denominan duplicados casi exactos o parciales (*fuzzy duplicates*) [27]. Por ejemplo, considérese un grafo de conocimiento inmobiliario con los nodos *Listing 1* y *Listing 2*, como se muestra en la Figura 1. Ambos nodos tienen *data properties* con valores idénticos (como el título "Hermosa propiedad en Av. Libertador al 400") y *object properties* que los vinculan a las mismas entidades (como `gr:hasPriceSpecification` que los une al precio y `foaf:maker` que referencia a su anunciante). Esto sugiere que ambos avisos, si bien son diferentes, publicitan la misma entidad subyacente: el inmueble en "Av Libertador al 400". Sin embargo, existen dos instancias de `RealEstate`, una para cada aviso. Del mismo modo, en la Figura 2, los nodos *Listing 1* y *Listing 2* muestran una diferencia en el precio al que se vinculan, pero siguen refiriéndose al mismo inmueble. Para representar este hecho, es importante establecer una relación `owl:sameAs` entre las instancias de los `RealEstate`, indicando que ambos representan la misma entidad.

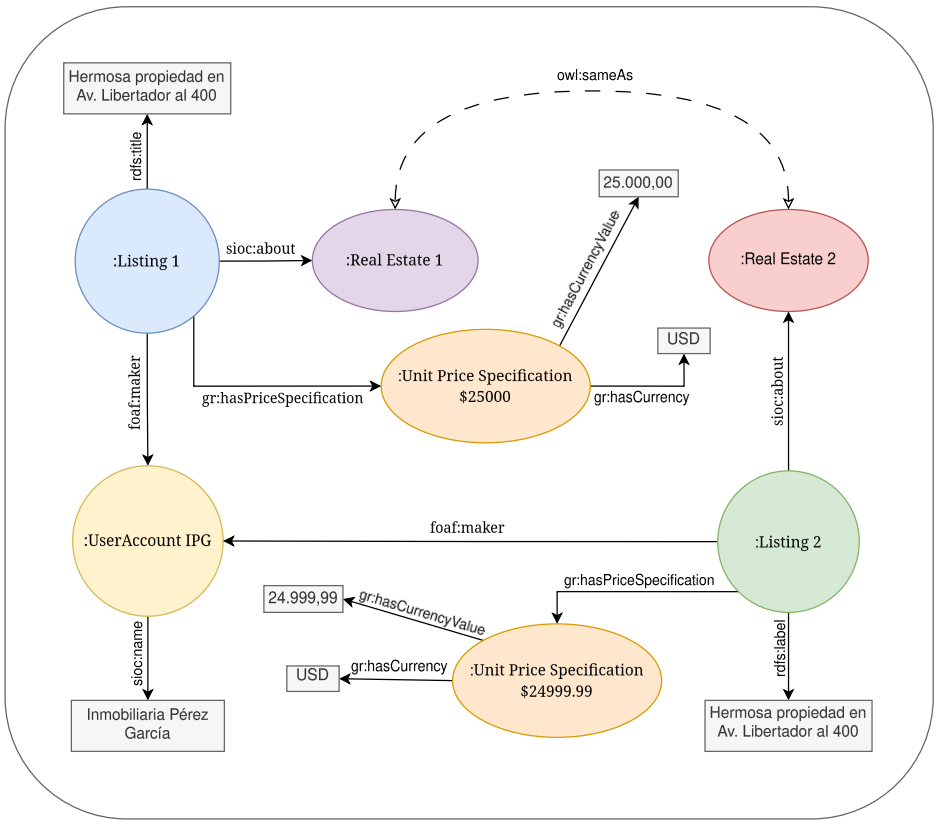

Figura 2. Duplicados parciales en un grafo de conocimiento.

El problema de la detección de duplicados ha recibido distintos nombres, como *record matching*, *record linkage*, deduplicación, *entity resolution*, e *instance matching*. Los autores varían su denominación según el contexto y las acciones que se toman para resolverlo. El problema de instance matching (IM) tiene como objetivo descubrir entidades duplicadas entre dos grafos de conocimiento. Tal y como lo definen Assi et al. [1] dados dos conjuntos de instancias $\mathcal{S}$ y $\mathcal{T}$ pertenecientes a dos KBs (KB_1 y KB_2), el objetivo del IM es descubrir el conjunto $\mathcal{M}$ de enlaces `owl:sameAs` según un criterio dado que no estén ya definidos en ninguno de los KBs. Su expresión formal se presenta en 1.

$$\begin{aligned}\mathcal{M} = \{(i_1, i_2) : i_1 \in \mathcal{S} \\ \wedge\, i_2 \in \mathcal{T} \\ \wedge\, i_1 \equiv i_2 \\ \wedge\, \langle i_1, owl : sameAs, i_2\rangle \notin KB_1 \cup KB_2\}\end{aligned} \tag{1}$$

La detección de duplicados en un grafo de conocimiento puede definirse como un problema de *instance matching* en el que el grafo es comparado consigo mismo, es decir, en el que $KG_1 = KG_2$, o $KB_1 = KB_2$ como se define en 1.

La detección de duplicados en grafos de conocimiento es compleja y depende del tamaño del grafo y de la naturaleza de los duplicados. Para resolver esta tarea de forma eficaz, suele dividirse en cuatro pasos: preprocesamiento, agrupamiento, comparación y agrupación. El preprocesamiento limpia y normaliza los datos para eliminar las diferencias sintácticas y de formato. El agrupamiento separa los datos en subconjuntos más pequeños para reducir el espacio de búsqueda. La comparación confronta los elementos de un grupo entre sí, también conocidos como candidatos a duplicados, y determina si se refieren a la misma entidad. La agrupación agrupa todos los duplicados que se refieren a la misma entidad.

En el contexto de este estudio, el grafo de conocimiento investigado incluye información sobre propiedades y sus anuncios. Los duplicados aparecen en el grafo como múltiples nodos que describen propiedades idénticas. Dado que varios anuncios pueden anunciar el mismo inmueble, se generan diferentes nodos para representar cada anuncio del mismo inmueble, y el mismo inmueble es descrito por tantos nodos como anuncios se hayan encontrado anunciándolo. Dentro de este marco, el estudio aborda el reto de identificar y vincular estas instancias como un problema de detección de duplicados.

5. Grafo de conocimiento de avisos inmobiliarios

En esta sección se presenta en detalle el grafo de conocimiento utilizado en el contexto del OVS, sobre el cual se implementa el sistema de detección de duplicados para los avisos inmobiliarios.

El grafo de conocimiento desempeña un papel fundamental en el OVS, sirviendo como repositorio organizado que captura anuncios provenientes de diversas fuentes y establece conexiones semánticas entre ellos. Los anuncios inmobiliarios en el grafo de conocimiento se recopilan desde una variedad de sitios

web y agencias inmobiliarias, generando así un conjunto de datos altamente heterogéneo. Esta diversidad puede plantear desafíos para la detección de duplicados debido a las disparidades en el formato y la estructura de la información.

5.1. Estructura

El grafo de conocimiento se almacena en RDF, un modelo estándar para codificar relaciones semánticas entre elementos de datos, de modo que estas relaciones puedan interpretarse computacionalmente. Las relaciones semánticas utilizadas para describir los avisos inmobiliarios, sus propiedades y características están especificadas en una ontología elaborada a partir de vocabularios estándar, como se detalla a continuación.

Ontología de avisos inmobiliarios. El esquema diseñado define las variables extraídas de los anuncios inmobiliarios de diversas páginas web, asegurando que todos los anuncios posean un conjunto coherente de propiedades. Esta ontología se emplea para transformar la información cruda en un grafo RDF, convirtiéndola en conocimiento interpretable por máquinas. Se derivan de vocabularios establecidos en la web y ontologías estándar, tales como GoodRelations [2] [18], schema.org [3] [16], FOAF [4] [6], y RealEstateCore [5] [17]. Estos recursos ofrecen definiciones y relaciones que se utilizan para estructurar el grafo de conocimiento. La ontología resultante, representada en la Figura 3, incluye las clases esenciales para reflejar los diversos aspectos del mercado inmobiliario. En la figura se distinguen las clases provenientes de cada ontología mediante un color particular y un prefijo específico en el formato `{prefix}:`, que es una abreviatura del dominio de cada ontología. Algunas entidades fueron creadas durante este trabajo para enlazar los conceptos de los distintos vocabularios. Para ellas, se utiliza el prefijo `:` (dos puntos). El propósito de cada clase se detalla a continuación:

- RealEstateCore
  - `rec:RealEstate`: Representación legal de un inmueble, entendido como uno o más terrenos/edificios.
  - `rec:Space`: Una parte contigua del mundo físico. Sus subclases se usan para representar tanto regiones (provincias, ciudades, barrios), como edificios y terrenos. Es semejante a la definición de `schema:Place`, que describe una entidad con una extensión física definida.
- SIOC
  - `sioc:Site`: Sitio web principal que alberga una plataforma en línea.
  - `sioc:Post`: Artículo publicado por un usuario en el foro de algún `sioc:Site`.
  - `sioc:UserAccount`: Cuenta online de un usuario en una plataforma. Con estas cuentas los usuarios pueden publicar un `sioc:Post`.

[2] GoodRelations: http://purl.org/goodrelations/
[3] Schema.org: https://schema.org/
[4] FOAF: http://xmlns.com/foaf/0.1/
[5] RealEstateCore: https://realestatecore.io/

- FOAF
  - `foaf:Agent`: Personas, organizaciones o grupos que realizan acciones. Quienes crean un `sioc:Post` desde una `sioc:UserAccount` para publicitar un `rec:RealEstate`, son considerados `foaf:Agent`. Pueden ser una persona particular o una agencia inmobiliaria.
- Good Relations
  - `gr:Offering`: Oferta pública con intención comercial realizada por un `foaf:Agent` (en este caso a través de un `sioc:Post`).
  - `gr:UnitPriceSpecification`: Conceptualización del precio pedido en una `gr:Offering`.
  - `gr:BusinessFunction`: Tipo de actividad relativo a una `gr:Offering`. Ej.: venta, alquiler, reparación, mantenimiento.
- Definidas para la ontología de avisos inmobiliarios de este trabajo
  - `:RealEstateListing`: Subclase de `sioc:Post` y `gr:Offering`. Oferta pública en forma de publicación online de un bien inmueble en venta o alquiler.
  - `:SizeSpecification`: Conceptualización del espacio ocupado por un inmueble en distintas medidas. Ej.: superficie total, superficie cubierta, superficie del terreno.
  - `:Feature`: Característica de un inmueble. Esta clase generalmente está poblada con la información no estructurada de los anuncios inmobiliarios que se encuentra en el formato clave: valor. La clave se almacena como un Literal en el `dc:title` (del vocabulario Dublin Core), y el valor de la misma forma en el `:has_value`. Un ejemplo de una característica que podría incluirse en esta clase es "Cantidad plantas: 5 o más".

Además, se utilizaron diferentes *object properties* para enlazar estas entidades entre sí y *data properties* para darles valores específicos para cada instancia. Algunas de ellas son:

- *Object properties*
  - `sioc:about`: Indica que un `:RealEstateListing` **trata de** un determinado `rec:RealEstate`.
  - `foaf:maker` y `foaf:made`: Definen a un `foaf:Agent` como el **creador de** un `:RealEstateListing`.
  - `schema:seller`: Similar a las anteriores, indica que un `rec:RealEstate` está siendo **vendido por** un `foaf:Agent`.
  - `rec:hasPart`: Usada para componer un `rec:Space` dentro de otro. En particular, se utiliza para decir que una habitación **es parte de** un edificio.
  - `rec:locatedIn`: Describe la ubicación geográfica de un edificio, representando que **se ubica en** un barrio, que a su vez **se ubica en** un partido y una provincia.
- *Data properties*
  - `schema:address`: Da la dirección física de un `rec:Space` como un `xsd:stirng`.
  - `gr:condition`: Describe la condición en la que se encuentra el inmueble según el aviso.

- `rdfs:label`: Aplicada sobre un `:RealEstateListing`, es el título del anuncio.
- `rdfs:comment`: Aplicada sobre un `:RealEstateListing`, es la descripción que escribió el anunciante en el aviso.
- `:is_brand_new`: Indica si el inmueble es a estrenar o no.
- `:is_finished`: Indica si el inmueble está en construcción o no.
- `:property_type`: Indica el tipo de propiedad que describe el anuncio. Ej.: casa, departamento, quinta.
- `gr:hasValue` y `gr:hasUnitOfMeasurement`: Indican el precio y la moneda en la que se ofrece el inmueble en ese aviso.
- `schema:latitude` y `schema:longitude`: Permiten vincular un `schema:Place` o semejante con sus coordenadas geográficas.

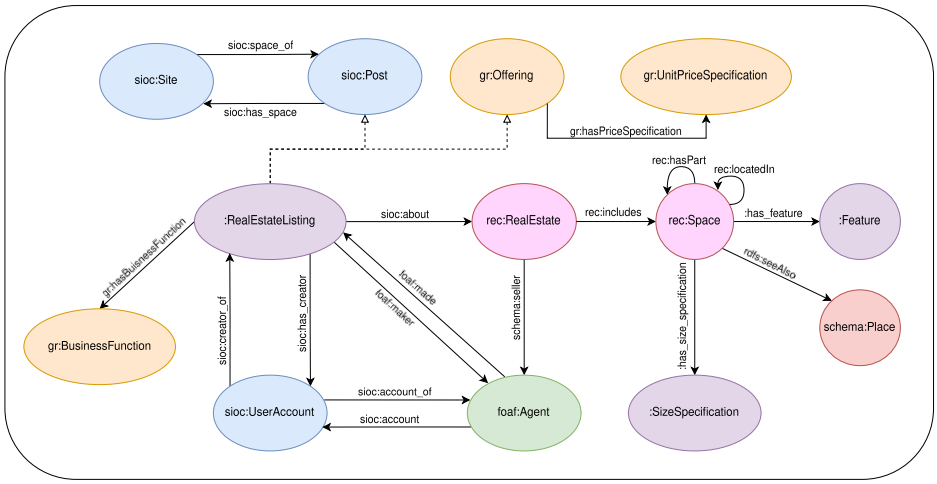

Figura 3. Definición de la ontología de avisos inmobiliarios.

La Figura 4 proporciona una representación visual de cómo la ontología se aplica para capturar relaciones semánticas entre diversos conceptos de los anuncios inmobiliarios. En el núcleo del gráfico se encuentra la instancia de `rec:RealEstate`, que encarna el concepto de un bien inmueble específico según la definición de RealEstateCore, identificado por el prefijo `rec:`. Esta instancia tiene *data properties* que detallan sus características, como `schema:address` que indica que la casa está ubicada en la calle Libertador al 400, y puede tener otros atributos definidos por `RealEstateCore` o la ontología específica desarrollada para este trabajo.

La instancia de `rec:RealEstate` se vincula con una agencia inmobiliaria particular, como `Inmobiliaria Pérez García`, representada como una instancia de `foaf:Agent`, mediante la propiedad `schema:seller`. Además, se establece a través de la propiedad `foaf:made` que el agente creó el aviso titulado “Hermosa

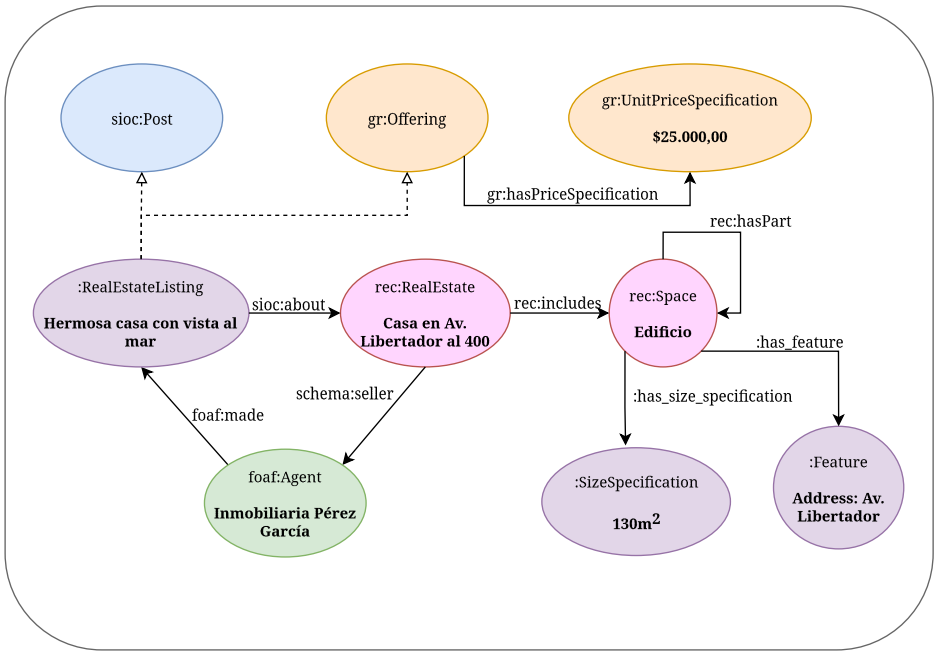

Figura 4. Ejemplo de un grafo de conocimiento mostrando la información de un aviso inmobiliario de acuerdo con la ontología.

casa con vista al mar", definido como una instancia de `:RealEstateListing`, que a su vez está asociada (`sioc:about`) con la instancia de `rec:RealEstate` previamente mencionada.

Asimismo, el bien inmueble se relaciona con una entidad que describe el espacio físico que ocupa (`rec:Space` a través de `rec:includes`), y esta entidad está vinculada con su dirección, representada por la entidad `:Feature`. Esta última agrupa instancias para diversas características genéricas de un inmueble. Cuando dos anuncios publicitan inmuebles en la misma dirección, ambos comparten una relación con esta instancia.

Patrones de creación de URIs. Los Unique Resource Identifiers (URI) sirven como identificadores únicos para las entidades del grafo de conocimiento. Es esencial que al construir y representar diversas entidades y sus relaciones se creen los URI correspondientes. Esto facilita la integración y consolidación de la información. Esencialmente, los URIs actúan como firmas digitales de los conceptos, permitiendo vincular y unificar los datos relacionados con una misma entidad, aunque sean de referencias y contextos distintos.

En el contexto de este grafo de conocimiento, que recopila información sobre el sector inmobiliario procedente de diversas plataformas y en distintos momentos, disponer de un método consistente para generar URIs permite con-

solidar toda la información relevante relativa a una entidad determinada, incluso si distintas instancias hacen referencia a ella en numerosas ocasiones. `Inmobiliaria Pérez García`, por ejemplo, podría haber publicado anuncios de distintas propiedades en distintos momentos. Por ejemplo, si hoy se menciona que `Inmobiliaria Pérez García` es la vendedora de una propiedad y hace dos semanas también fue citada como la vendedora de otra propiedad, el uso de URIs permite establecer que ambas instancias se refieren a la misma entidad de agencia.

Para garantizar que una única entidad representa a esta agencia, en lugar de crear múltiples entidades separadas, se utilizan las URIs. Al asignar una URI consistente a `Inmobiliaria Pérez García`, se busca asegurar que la próxima vez que aparezca en los datos, se le asigne el mismo URI. Al asignar y utilizar el mismo URI de manera coherente, se garantiza la continuidad y consistencia en la identificación de la entidad a lo largo del tiempo y en diferentes contextos. De este modo, el URI actúa como un identificador persistente y unívoco, facilitando la fusión de información relacionada con la agencia, independientemente de dónde o cuándo aparezca en los datos.

A continuación, se describirán los URIs generados para cada tipo de entidad, usando *URI Templates* de nivel 1 según se definen en el RFC 6570 [13]. Una vez más, se abreviarán los dominios de las ontologías alojadas en la Web usando prefijos en el formato `prefix:`, y se utilizará el prefijo `:` para indicar los URIs de la ontología presentada en este trabajo.

Para los `:RealEstateListing` se generan URIs de la siguiente forma:

```
:listing_{site_id}_{listing_id}
```

donde `site_id` representa el identificador de la plataforma de anuncios inmobiliarios de la que proviene el aviso, y `listing_id` es el identificador único asignado por esa plataforma a ese aviso específico.

La asignación de URIs para los anuncios inmobiliarios sigue el formato:

```
:listing_{site_id}_{listing_id}
```

donde `{site_id}` representa el identificador de la plataforma de anuncios inmobiliarios de la que proviene el aviso, y `{listing_id}` es el identificador único asignado por esa plataforma al anuncio específico. En casos donde `{site_id}` o `{listing_id}` están ausentes, se utiliza un valor incremental para mantener la singularidad. Específicamente, para los anuncios sin un identificador, la URI se convierte en:

```
:listing_{timestamp}_{incremental}
```

donde `{timestamp}` indica el momento de asignación de la URI, y `{incremental}` es un número natural que identifica la entidad de manera incremental. La Figura 5 ilustra cómo pueden lucir las instancias del grafo de conocimiento al adoptar estos patrones.

De manera similar, para las agencias inmobiliarias, la URI toma la forma `:agent_{name}` utilizando el nombre de la agencia o agente. En caso de ausencia

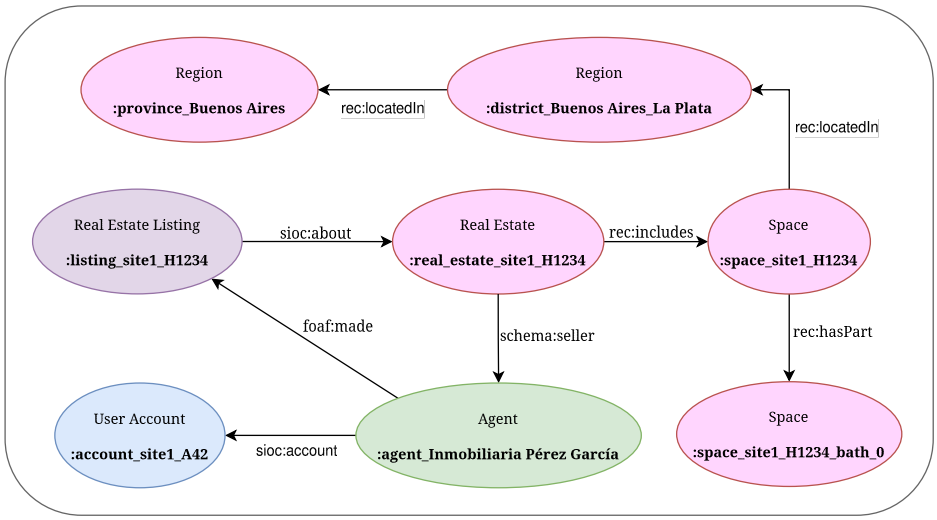

Figura 5. Entidades en el grafo de conocimiento con URIs acordes a los patrones definidos.

de nombre, se omite el nodo del grafo. La URI para cuentas de usuario sigue el patrón:

```
:account_{site}_{advertiser_id}
```

donde `{site}` representa el identificador de la plataforma (por ejemplo, `site1`, `site2`), y `{advertiser_id}` es el identificador único de cuenta asignado por la plataforma.

Las entidades inmobiliarias se identifican con URIs como:

```
:real_estate_{site}_{listing_id}
```

reflejando la estructura de los anuncios. Esto establece una clara conexión entre los anuncios inmobiliarios y las propiedades que promocionan. En situaciones en las que falta información esencial, se aplica un valor incremental para garantizar la distinción de URIs.

Las URIs para instancias de `Space` adoptan el formato:

```
:space_{site}_{listing_id}container
```

mientras que las URIs para barrios, distritos y provincias se formulan respectivamente de la siguiente forma:

```
:neighborhood_{province}_{district}_{neighborhood}
:district_{province}_{district}
:province_{province}
```

Aquí, `{province}`, `{district}` y `{neighborhood}` representan sus respectivos nombres (ver Figura 5). Las instancias que carecen de información necesaria se omiten del grafo. Finalmente, las instancias de `Room` dentro de un `Space` se identifican con URIs como:

`:{space}_{room}_{incremental}`

donde `{space}` designa el espacio contenedor, `{room}` especifica el tipo de habitación (por ejemplo, baño, dormitorio) y `{incremental}` indica el conteo incremental de la habitación dentro del espacio.

5.2. Extracción de conocimiento

La selección de las fuentes de datos es un paso crucial en el proceso de recopilación de datos. Para ello, un equipo de expertos en el dominio evaluó detenidamente varios sitios web inmobiliarios destacados de la región. Tras la evaluación, se consideró que tres sitios web eran los más representativos del mercado inmobiliario. Esta selección garantizó que los datos recogidos fueran lo suficientemente sólidos para el análisis estadístico. Además, los llevó a considerar que si bien la inclusión de más sitios web no aumentaría la calidad ni la cantidad de la información inmobiliaria, la eliminación o sustitución de cualquiera de ellos sí reduciría significativamente la cantidad de conocimientos recuperados.

Dado que no todos los sitios web seleccionados ofrecen una API para la recuperación de datos, se optó por utilizar una solución de *web scraping*. El *web scraping*, también conocido como *web extraction* o *harvesting*, es una técnica para extraer datos de la World Wide Web y guardarlos en una base de datos para su posterior recuperación o análisis [34]. Por su parte, un *web crawler*, *spider*, o simplemente araña, es un agente especializado en la descarga masiva de sitios web [29], [31]. El *web scraper* está desarrollado con Python, utilizando el framework Scrapy [6]. La arquitectura del *scraper* comprende tres arañas distintas, cada una adaptada para extraer datos de un sitio web específico. Este diseño modular garantiza la escalabilidad, permitiendo una fácil integración de sitios adicionales mediante la creación y vinculación de nuevas arañas al sistema.

Para evitar sobrecargar los servidores de las plataformas y minimizar las posibles pérdidas, el *scraper* incorpora un tiempo de retraso entre las peticiones enviadas a cada sitio web. Además, ejecutar todas las arañas concurrentemente optimiza la eficiencia computacional, permitiendo procesar los avisos mientras se gestionan otras peticiones.

Dada la diversidad de fuentes de datos, se hizo evidente la necesidad de normalizar el vocabulario utilizado por cada sitio web para representar el conocimiento. Por este motivo, se estableció una colaboración con los expertos del dominio con el fin de identificar las variables críticas que debían extraerse de cada sitio web, junto con sus contrapartes en las otras plataformas. Esto dio como resultado el desarrollo de un vocabulario estandarizado que sintetiza la

[6] Scrapy: https://scrapy.org/

información a recolectar de cada anuncio, independientemente de las variaciones en la terminología utilizada por diferentes páginas. Aunque algunas páginas pueden carecer de atributos de datos específicos o permitir valores nulos, dando a las agencias inmobiliarias la opción de excluir detalles, estos casos son comparativamente raros y no obstaculizan el proceso global.

Por último, el *scraper* implementa un módulo que permite almacenar el conocimiento extraído por las distintas arañas simultáneamente en la misma base. Este método elimina la necesidad de generar bases de conocimiento distintas para cada plataforma o para cada araña, lo que simplifica el proceso de consolidación del conocimiento.

5.3. Conversión del formato e inferencia de conocimiento

Como resultado del proceso de scraping se obtiene una base de datos que contiene un amplio repositorio de datos de anuncios inmobiliarios. En esta sección se ahonda sobre el marco metodológico empleado para la construcción del grafo de conocimiento mediante un *script* de Python, que culmina en la generación de un archivo RDF.

El paso inicial del *script* implica el uso de la biblioteca RDFLib [24] para modelar una estructura de grafos, cargando la información ontológica definida. Luego, procesa los datos extraídos de manera completamente independiente para cada anuncio. Es fundamental destacar que, en esta fase, se asume que cada anuncio publicita una propiedad única. En otras palabras, se crea un nodo de `rec:RealEstate` distinto para cada aviso. El proceso de deduplicación, en el que se vincularán estos nodos, se reserva para una fase posterior de postprocesamiento una vez construido el grafo.

Para cada anuncio, el *script* genera un grafo con toda su información, siguiendo las especificaciones de la ontología, para luego agregar estos subgrafos a un único grafo común que contendrá la totalidad del conocimiento. El módulo encargado de generar estos subgrafos comienza por anonimizar los datos sensibles, reemplazando los dominios de los sitios con etiquetas genéricas como "sitio1", "sitio2", etc., como se explicó en la Sección 5.1. Del mismo modo, se modifican los ID de los anuncios que podrían revelar información específica del sitio.

El módulo de generación crea las entidades principales que componen el grafo conforme a las especificaciones de la ontología. Estas entidades incluyen `:RealEstateListing`, `foaf:Agent`, `rec:RealEstate`, así como aquellas relacionadas, como `sioc:UserAccount` asociada a `foaf:Agent` y los `rec:Space` vinculados a `rec:RealEstate`. Es importante destacar que a cada nodo creado se le asigna un URI según las directrices establecidas en la sección correspondiente. Cuando no se puede garantizar la unicidad de alguno de los URI, se le agregan valores como la fecha y hora de creación, y un número incremental. De esta forma se busca evitar que un mismo nodo tenga el conocimiento de dos avisos diferentes (lo que sucedería si compartieran URI).

A medida que se generan los subgrafos, el *script* se encarga de combinarlos de manera iterativa en una sola estructura que recopila todo el conocimiento extraído. De esta manera, se concluye con una única base de conocimiento en

formato RDF que puede contener entidades duplicadas que necesiten tratamiento.

6. Enfoque Bayesiano para detectar duplicados

En la siguiente sección se presenta la estrategia bayesiana empleada para la detección de duplicados, junto con la herramienta seleccionada para implementar el pipeline. Cada componente del proceso de clasificación será descrito en detalle, especificando cómo contribuye a resolver el problema.

Muchas de las herramientas existentes para la comparación de entidades en la detección de duplicados suelen basarse en una única función de similitud y tratan todos los atributos del mismo modo, como strings o vectores de caracteres. Sin embargo, el enfoque evaluado propone apartarse de este método convencional utilizando un proceso de comparación segmentado. Esta estrategia aboga por segmentar la comparación en función de los atributos de los anuncios evaluados, asignando pesos distintos a cada segmento para obtener un resultado de coincidencia global.

Para determinar si dos anuncios son duplicados, se comparan sus atributos respectivos. Por ejemplo, sean considerados dos anuncios con atributos como "ubicación", "número de habitaciones", y "precio". Primero, se compara la ubicación de ambos anuncios, luego se compara el número de habitaciones y, finalmente, se compara el precio. Para cada comparación, se calcula una puntuación de similitud, que se asigna a ese atributo específico. Estas puntuaciones se combinan utilizando el Teorema de Bayes para generar una puntuación de similitud global que representa la similitud entre los dos registros.

Este enfoque permite calcular la probabilidad de que los registros sean duplicados. Al comparar esta probabilidad con un umbral de similitud especificado por el usuario, se puede determinar si los registros coinciden o no.

6.1. Configuración y Estrategias de Comparación

La detección de duplicados con este enfoque bayesiano implica el uso de diversas funciones de similitud adaptadas a la naturaleza de los atributos comparados. Se seleccionan funciones específicas según el tipo de atributo para lograr una correspondencia precisa. Por ejemplo, al comparar cadenas extensas como las descripciones de los anuncios, se utiliza la métrica de distancia de Levenshtein, que mide el número de cambios necesarios para transformar un texto en otro. En el contexto del grafo de conocimiento inmobiliario, cada propiedad se compone de múltiples atributos semánticos, como la descripción, el número de dormitorios y la ubicación. El enfoque bayesiano emplea estas comparaciones sintácticas para evaluar la probabilidad de equivalencia semántica.

Un punto clave en la configuración de la estrategia de deduplicación es el uso de Lucene [7] durante la etapa de agrupamiento, con la que se ejecutan consultas

[7] Lucene: https://lucene.apache.org/

simples y eficientes para identificar posibles duplicados y descartar a la mayor cantidad de candidatos improbables. Las funciones de búsqueda e indexación de Lucene posibilitan que la estrategia realice particiones y búsquedas eficientes en los datos, facilitando la identificación de registros con información textual similar.

En la fase de preprocesamiento, se aplicaron procesos estándar de curado, que incluyeron la eliminación de espacios redundantes y la conversión del texto a minúsculas, entre otras técnicas. Esta etapa tiene como objetivo normalizar los datos y mejorar los resultados del análisis.

Durante el proceso de matcheo, se emplearon comparadores especializados para abordar diferentes tipos de datos. Para cadenas de texto largas, como descripciones o direcciones, se utilizó el comparador Levenshtein para medir la similitud entre los registros. En el caso de cadenas más cortas, como los nombres de los lugares, se evaluaron mediante el comparador Jaro-Winkler. Para atributos numéricos, como la superficie y el número de habitaciones, se utilizaron comparadores numéricos específicos. Estos comparadores calculan el cociente entre el número mayor y el menor, teniendo en cuenta la proporcionalidad de los atributos.

Además, se utilizó un comparador de geolocalizaciones para medir la distancia entre dos coordenadas. Si la distancia superaba los 100 metros (lo que indicaba una separación espacial significativa), se consideraba que los registros estaban completamente alejados. Los valores específicos de la configuración se describen en el Cuadro 1, que también muestra el rango del umbral de similitud seleccionado para cada variable. Estos valores representan la probabilidad de que un par de registros sean duplicados, cuando los valores de ese atributo son exactamente iguales o completamente diferentes, según el comparador utilizado. Por ejemplo, dado que los títulos de dos anuncios inmobiliarios son exactamente iguales, la probabilidad de que los anuncios se refieran a la misma entidad es del 70 %. Del mismo modo, si los títulos son completamente diferentes, esa probabilidad se reduce al 19 %. Las probabilidades calculadas para cada atributo son las que se reducen a un único número mediante el Teorema de Bayes.

7. Métricas

En esta sección se describen las métricas utilizadas para evaluar el rendimiento del algoritmo de detección de duplicados. Se explicarán las métricas proporcionadas, incluyendo los conocimientos que aportan y cómo se calculan.

La detección de duplicados es un problema de clasificación binaria, en el que el algoritmo recibe un par de registros y debe clasificarlos como coincidentes o no coincidentes. Cada clasificación realizada por el algoritmo cae en una de las siguientes categorías:

- Verdadero positivo (TP): El algoritmo identifica correctamente una coincidencia entre dos entidades.
- Falso positivo (FP): El algoritmo identifica incorrectamente una coincidencia entre dos entidades.

Cuadro 1. Configuración usada para detectar duplicados en el grafo de conocimiento.

Atributo	Comparador	Umbral inferior de similitud	Umbral superior de similitud
Dirección	Levenshtein	0.09	0.90
Antiguedad	Exacto	0.07	0.50
Baños	Exacto	0.17	0.51
Dormitorios	Exacto	0.10	0.60
Garages	Exacto	0.10	0.60
Ambientes	Exacto	0.10	0.61
Coordenadas	Geoposición	0.20	0.80
Superficie cubierta	Numérico	0.09	0.55
Descripción	Levenshtein	0.39	0.70
Partido	Jaro-Winkler	0.39	0.51
Superficie del terreno	Numérico	0.07	0.60
Expensas	Numérico	0.05	0.60
Precio	Numérico	0.05	0.80
Tipo de propiedad	Exacto	0.10	0.56
Título	Levenshtein	0.19	0.90
Superficie total	Numérico	0.10	0.67

- Verdadero negativo (TN): El algoritmo identifica correctamente que dos registros no coinciden.
- Falso negativo (FN): El algoritmo identifica incorrectamente que dos registros no coinciden.

El rendimiento global del algoritmo depende del número de coincidencias que haya dentro de cada categoría.

Tres métricas serán usadas para la evaluación: precisión, exhaustividad y Valor-F [12]. La precisión mide la exactitud del algoritmo a la hora de clasificar un par de registros como coincidentes, computando los resultados positivos verdaderos entre todos los resultados positivos. Se define como se muestra en la Ecuación 2.

$$\text{Precisión} = \frac{TP}{TP + FP} \tag{2}$$

Por otro lado, la exhaustividad indica la proporción de verdaderos positivos entre todos los positivos reales. Su objetivo es describir la capacidad del algoritmo para detectar pares duplicados. Se define como se indica en la ecuación 3.

$$\text{Exhaustividad} = \frac{TP}{TP + FN} \tag{3}$$

Por ejemplo, si una herramienta de deduplicación registra una exhaustividad de 0,3 y precisión de 0,9, significa que identifica correctamente el 90 % de los registros que etiqueta como duplicados, pero solo identifica el 30 % de todos los registros duplicados de la base de datos.

El Valor-F es una forma de combinar la precisión y la recuperación en una única métrica, definida como se muestra en la ecuación 4.

$$\text{Valor} - \text{F} = 2 \cdot \frac{\text{Precisión} \cdot \text{Exhaustividad}}{\text{Precisión} + \text{Exhaustividad}} \tag{4}$$

8. Evaluación

Esta sección de la investigación se evalúa el rendimiento de la estrategia de detección de duplicados presentada en la Sección 6 sobre una base de conocimientos de anuncios inmobiliarios. En primer lugar, se explicará la creación de un *ground truth* gracias al etiquetado manual de los expertos del dominio. A continuación, se presentará la herramienta utilizada para evaluar el enfoque bayesiano, explicando los detalles específicos de diseño e implementación.

8.1. Creación del Ground Truth

El término *ground truth* se refiere a un conjunto de datos que contiene información veraz, obtenida mediante evidencia empírica y utilizado como estándar para evaluar diversos algoritmos. En este estudio se generó un *ground truth* específico para evaluar el rendimiento de la estrategia de detección de duplicados presentada en la Sección 6. Este conjunto de datos consta de dos archivos distintos: una base de conocimiento que contiene toda la información sobre cada anuncio inmobiliario y un archivo de enlaces `owl:sameAs` que especifica qué anuncios son duplicados entre sí.

Para crear este conjunto de datos, un equipo de expertos utilizó la herramienta de *web scraping* presentada en la Sección 5.2 para generar una base de conocimiento de 623840 anuncios inmobiliarios. Con el fin de minimizar el riesgo de omitir duplicados, dividieron el conjunto de datos en subconjuntos más pequeños basándose en la proximidad de los inmuebles, determinada por la latitud y longitud publicada en los anuncios. Los expertos revisaron manualmente cada subconjunto para identificar duplicados, comparando diversos atributos como el título, la descripción, el precio, el número de dormitorios, la antigüedad, el tipo de propiedad y la dirección. Registraron los duplicados encontrados en una planilla, resultando en 3688 filas. Cada fila representa un conjunto de anuncios que son duplicados entre sí, y cada conjunto puede contener dos o más anuncios, ya que varios anuncios pueden representar la misma propiedad. Luego, los expertos eliminaron de la base de conocimiento los anuncios que no habían sido marcados como duplicados en la hoja de cálculo, reduciendo la base a 9333 anuncios.

Para crear el *ground truth*, se implementó un *script* que aleatoriza las filas de la hoja de cálculo que contiene los ID de los anuncios duplicados y la divide por la mitad. La primera mitad representaba los anuncios duplicados, mientras que la otra mitad representaba los no duplicados. La primera mitad se utilizó para construir una nueva base de conocimiento con los anuncios que se sabía que estaban duplicados, guardando en un archivo los enlaces `owl:sameAs` que representaban qué anuncios eran duplicados entre sí.

Para la segunda mitad, se seleccionó aleatoriamente un anuncio de cada conjunto para que formara parte de la nueva base de conocimiento, mientras que los demás anuncios se eliminaron. De esta manera, se garantizaba que los anuncios seleccionados no tuvieran duplicados conocidos en la nueva base de datos.

A continuación, se proporciona un ejemplo para explicar la metodología de construcción de conjuntos de datos. Supongamos que tenemos una planilla como la que se muestra en el Cuadro 2, con cuatro filas de anuncios que describen propiedades diferentes, etiquetadas como A, B, C y D. Las propiedades A y D están descritas por dos anuncios cada una, mientras que la propiedad B tiene cuatro anuncios y la C tiene tres.

Cuadro 2. Planilla de ejemplo con avisos duplicados.

Propiedad	ID de avisos
A	A_1, A_2
B	B_1, B_2, B_3, B_4
C	C_1, C_2, C_3
D	D_1, D_2

El proceso de construcción del conjunto de datos consiste en aleatorizar la planilla y dividirla en dos mitades: una para los duplicados y otra para los no duplicados. El Cuadro 3 ilustra este escenario.

Cuadro 3. Propiedades de ejemplo con los ID de las propiedades que las describen.

Propiedad	ID de avisos	Grupo
C	C_1, C_2, C_3	Duplicados
B	B_1, B_2, B_3, B_4	Duplicados
D	D_1, D_2	No-duplicados
A	A_1, A_2	No-duplicados

En la primera mitad, que incluye las propiedades C y B, cada anuncio se trata como un duplicado de los demás anuncios de su propiedad respectiva. En otras palabras, cada anuncio de la propiedad C se considera un duplicado de los demás anuncios de la propiedad C, y lo mismo ocurre para la propiedad B. Esto resulta en una nueva tabla que contiene los pares duplicados de las propiedades C y B, como se muestra en el Cuadro 4. Luego, la información de todos estos anuncios se guarda en un nuevo archivo que corresponde al *ground truth*.

En la segunda mitad, se elige un único anuncio al azar de cada propiedad (por ejemplo, A_2 para la propiedad A y D_1 para la propiedad D). Los datos de los anuncios seleccionados se añaden al *ground truth*, mientras que los anuncios restantes (A_1 y D_2) no se incluyen. Al final de este proceso, la propiedad C tiene tres pares de duplicados conocidos, mientras que la propiedad B tiene seis. En cambio, los inmuebles A y D no tienen ninguno.

Cuadro 4. Pares de avisos duplicados.

Primer anuncio	Segundo anuncio
C_1	C_2
C_2	C_3
C_1	C_3
B_1	B_2
B_2	B_3
B_3	B_4
B_1	B_3
B_1	B_4
B_2	B_4

Una vez completado este proceso, los archivos resultantes contienen información sobre múltiples anuncios, reflejando las complejidades del mundo real, que incluyen tanto múltiples anuncios de una misma propiedad (duplicados) como propiedades representadas por un único anuncio (sin duplicados). Este *ground truth* identifica qué anuncios representan duplicados y cuáles no, y, por lo tanto, sirve como un conjunto de datos confiable para evaluar una estrategia de detección de duplicados dentro de este dominio.

8.2. Duke

Duke [8] es un motor potente y versátil de *entity resolution* desarrollado en Java. Aunque se desarrolló como parte del sistema de archivo de documentos Sesam [15], Duke es independiente del dominio, lo que significa que puede utilizarse con datos de cualquier índole y estructura.

A pesar de que Duke se refiere a las entidades de datos como "registros" durante el proceso de deduplicación, su salida consiste en enlaces que representan conexiones `owl:sameAs` entre duplicados. El proceso de detección de duplicados de Duke consta de tres pasos, que se describen a continuación.

En primer lugar, Duke carga cada registro del grafo de conocimiento y preprocesa sus atributos utilizando `Cleaner`s Estos objetos Java, que implementan la interfaz `Cleaner`, normalizan los datos mediante operaciones como la eliminación de espacios adicionales, la conversión de mayúsculas a minúsculas y la eliminación de caracteres superfluos.

Luego, Duke utiliza una implementación de base de datos para indexar cada registro. Esto le permite llamar a la función `findCandidateMatches(record)` para cada uno de los registros y emparejarlos con su par más probable según el criterio de indexación.

Después de la etapa de agrupamiento, Duke empareja cada par de registros comparando sus atributos mediante un comparador. La interfaz de comparación (`Comparator`) de Duke permite la definición de funciones de comparación personalizadas que asignan puntuaciones de similitud a los atributos de los registros.

[8] Duke: https://github.com/larsga/Duke/

Estas puntuaciones se agregan para obtener una puntuación global que representa la similitud entre dos registros. Duke incluye varios comparadores integrados, como medidas de distancia de cadenas como Jaro-Winkler y Levenshtein, y medidas basadas en tokens como el coeficiente de Dice y el índice de Jaccard. Al igual que con las otras interfaces, la interfaz de comparación de Duke se puede aprovechar para incluir funciones de comparación personalizadas que consideren conocimientos y características específicas del dominio.

Para determinar si dos registros son duplicados, Duke calcula una puntuación de similitud entre ellos utilizando el enfoque descrito para el emparejamiento. Esta puntuación de similitud representa la probabilidad de que los registros sean duplicados. Posteriormente, Duke compara la probabilidad resultante con un umbral definido por el usuario. Si la probabilidad supera el umbral, los registros se clasifican como duplicados.

9. Resultados

La estrategia de detección de duplicados se evaluó utilizando el *ground truth* descrito en la Sección 8.1. La salida de Duke se comparó con el archivo de enlaces del *ground truth*, y la precisión, la exhaustividad, y el Valor-F se utilizaron como métricas de rendimiento.

El *ground truth* contenía 6543 registros, y la estrategia logró encontrar 3139 de 4455 enlaces correctos, lo que resultó en una exhaustividad de aproximadamente 70,4 %. Sin embargo, es importante señalar que la estrategia también encontró 1554 enlaces que no estaban en el *ground truth*, lo que resultó en una precisión de solo el 66,8 %. En general, esto lleva a un Valor-F de 68,6 %.

Este hallazgo es interesante y podría atribuirse a varios factores. Por ejemplo, si los anuncios inmobiliarios fueron publicados por la misma agencia, es posible que hayan utilizado plantillas de texto para la descripción y el título del anuncio, lo que lleva a un mayor número de falsos positivos por tener más contenido en común. En este caso, podría valer la pena evaluar una configuración diferente con pesos más bajos en estas variables. Otra opción podría ser utilizar una estrategia de deduplicación más flexible y que no dependa únicamente de la clasificación bayesiana, que asume que las probabilidades dadas por cada variable son independientes. Por ejemplo, dicha estrategia podría asignar un peso menor a la variable de descripción cuando los anuncios son publicados por la misma agencia, ya que es probable que sus avisos tengan descripciones similares.

En cuanto a las coordenadas de los listados, se descubrió que las agencias inmobiliarias pueden no utilizar la latitud y longitud reales de la propiedad, marcando, en cambio, un punto cercano, o utilizando la ubicación de la agencia misma. Esto puede llevar a que muchos registros tengan valores de coordenadas similares, aunque se refieran a propiedades diferentes. Por lo tanto, el hecho de que dos coordenadas sean diferentes o iguales no descarta ni confirma necesariamente una coincidencia, respectivamente.

Cabe destacar que, si bien el *ground truth* sirve como una muestra representativa del grafo de conocimiento real, identificar tantos duplicados puede ser

desafiante. Los expertos humanos pueden pasar por alto algunos duplicados, y el proceso de generación del *ground truth* en sí mismo puede introducir errores. Por lo tanto, se requiere un refinamiento continuo de la estrategia de detección de duplicados para mejorar la precisión de los resultados y adaptarse a la naturaleza evolutiva de los datos inmobiliarios.

Cuando la herramienta de detección de duplicados se ejecutó en un grafo de conocimiento completo que comprende 79100 listados, completó la tarea en 3 horas y 44 minutos, utilizando un procesador Intel Core i5 de 7ª generación con 4 núcleos y 8 GB de RAM.

10. Conclusiones y Trabajos Futuros

Este estudio evaluó una estrategia de detección de duplicados aplicada a un grafo de conocimiento inmobiliario compilado a partir de diversas fuentes. El enfoque implicó calcular una puntuación de probabilidad que indica la chance de que dos registros sean duplicados, lo que permitió la identificación de enlaces `owl:sameAs` implícitos entre ellos. La evaluación de este enfoque resultó en una precisión del 66,8 %, una exhaustividad del 70,4 % y un Valor-F del 68,6 %. Al segmentar el proceso de comparación de registros según los atributos considerados y asignar pesos a cada segmento, la estrategia abordó eficazmente el desafío de la detección de duplicados. Además, el enfoque demostró su capacidad para limpiar el grafo de conocimiento mediante la eliminación de datos que generaran ruido, mejorando así la calidad y confiabilidad de la información inmobiliaria disponible para el análisis dentro del dominio. En el marco del OVS, reducir las inconsistencias en la base de conocimiento implicará una mejora directa en los resultados de las estadísticas obtenibles, dado que podrá realizar los cálculos basándose en un muestreo del mercado inmobiliario real, y no uno sesgado por la cantidad de avisos referidos a los mismos inmuebles. Además, la deduplicación permitirá a los analistas revisar las contradicciones en los datos y ayudará a refinar el proceso de extracción y curado de información a futuro.

La estrategia se evaluó mediante el uso de un *ground truth* creado por expertos en el dominio, que consta de un grafo de conocimiento de avisos inmobiliarios con entidades duplicadas y no duplicadas, así como un archivo de enlaces que identifica los enlaces `owl:sameAs` faltantes en el conjunto de datos.

En el futuro, se explorarán configuraciones alternativas para mejorar la precisión general del enfoque presentado y para identificar más duplicados. Además, se experimentarán con otras técnicas para detectar avisos inmobiliarios duplicados, como análisis estructural de grafos, ventanas deslizantes, *graph embedding* y soluciones impulsadas por aprendizaje automático.

Referencias

1. Assi, A., Mcheick, H., Dhifli, W.: Data linking over RDF knowledge graphs: A survey. Concurrency and Computation: Practice and Experience **32**(19) (2020). https://doi.org/10.1002/cpe.5746, https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.5746, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.5746

2. Barlaug, N., Gulla, J.A.: Neural Networks for Entity Matching: A Survey. ACM Transactions on Knowledge Discovery from Data **15**(3), 1–37 (Jun 2021). https://doi.org/10.1145/3442200, https://dl.acm.org/doi/10.1145/3442200
3. Batini, C., Scannapieca, M.: Introduction to data quality. Data Quality: Concepts, Methodologies and Techniques pp. 1–18 (2006)
4. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web: A New Form of Web Content That is Meaningful to Computers Will Unleash a Revolution of New Possibilities. Scientific American p. 4 (May 2001)
5. Binette, O., Steorts, R.C.: (Almost) all of entity resolution. Science Advances **8**(12), eabi8021 (Mar 2022). https://doi.org/10.1126/sciadv.abi8021, https://www.science.org/doi/10.1126/sciadv.abi8021
6. Brickley, A., Constabaris, D., Graves, M.: FOAF: Connecting People on the Semantic Web. In: Knitting the Semantic Web. Routledge (2007), num Pages: 12
7. Dong, X.L., Berti-Equille, L., Srivastava, D.: Integrating conflicting data: the role of source dependence. Proceedings of the VLDB Endowment **2**(1), 550–561 (Aug 2009). https://doi.org/10.14778/1687627.1687690, https://dl.acm.org/doi/10.14778/1687627.1687690
8. Elmagarmid, A., Ipeirotis, P., Verykios, V.: Duplicate Record Detection: A Survey. Knowledge and Data Engineering, IEEE Transactions on **19**, 1–16 (Feb 2007). https://doi.org/10.1109/TKDE.2007.250581
9. Enamorado, T., Fifield, B., Imai, K.: Using a Probabilistic Model to Assist Merging of Large-scale Administrative Records. American Political Science Review (2018)
10. Fellegi, I.P., Sunter, A.B.: A Theory for Record Linkage. Journal of the American Statistical Association **64**(328), 1183–1210 (Dec 1969). https://doi.org/10.1080/01621459.1969.10501049, http://www.tandfonline.com/doi/abs/10.1080/01621459.1969.10501049
11. Fensel, D., Şimşek, U., Angele, K., Huaman, E., Kärle, E., Panasiuk, O., Toma, I., Umbrich, J., Wahler, A.: Knowledge Graphs: Methodology, Tools and Selected Use Cases. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-37439-6, http://link.springer.com/10.1007/978-3-030-37439-6
12. Ferrara, A., Lorusso, D., Montanelli, S., Varese, G.: Towards a Benchmark for Instance Matching. The 7th International Semantic Web Conference p. 13 (Jan 2008)
13. Fielding, R.T., Nottingham, M., Orchard, D., Gregorio, J., Hadley, M.: URI Template. Request for Comments RFC 6570, Internet Engineering Task Force (Mar 2012). https://doi.org/10.17487/RFC6570, https://www.rfc-editor.org/info/rfc6570, num Pages: 34
14. Garshol, L.M.: Duke (Apr 2011), https://github.com/larsga/Duke
15. Garshol, L.M., Borge, A.: Hafslund Sesam – An Archive on Semantics. In: Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G., Cimiano, P., Corcho, O., Presutti, V., Hollink, L., Rudolph, S. (eds.) The Semantic Web: Semantics and Big Data, vol. 7882, pp. 578–592. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38288-8_39, http://link.springer.com/10.1007/978-3-642-38288-8_39, series Title: Lecture Notes in Computer Science
16. Guha, R.: Introducing schema. org: Search engines come together for a richer web. Google Official Blog (2011)

17. Hammar, K., Wallin, E.O., Karlberg, P., Hälleberg, D.: The RealEstateCore Ontology. In: Ghidini, C., Hartig, O., Maleshkova, M., Svátek, V., Cruz, I., Hogan, A., Song, J., Lefrançois, M., Gandon, F. (eds.) The Semantic Web – ISWC 2019, vol. 11779, pp. 130–145. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-30796-7_9, https://link.springer.com/10.1007/978-3-030-30796-7_9, series Title: Lecture Notes in Computer Science
18. Hepp, M.: GoodRelations: An Ontology for Describing Products and Services Offers on the Web. In: Gangemi, A., Euzenat, J. (eds.) Knowledge Engineering: Practice and Patterns. pp. 329–346. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87696-0_29
19. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman & Hall - CRC Press (Aug 2009). https://doi.org/10.1201/9781420090512, journal Abbreviation: Foundations of Semantic Web Technologies Publication Title: Foundations of Semantic Web Technologies
20. Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G.D., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., Ngomo, A.C.N., Polleres, A., Rashid, S.M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., Zimmermann, A.: Knowledge Graphs. ACM Computing Surveys **54**(4), 1–37 (May 2022). https://doi.org/10.1145/3447772, https://dl.acm.org/doi/10.1145/3447772
21. Huaman, E., Kärle, E., Fensel, D.: Duplication detection in knowledge graphs: Literature and tools. CoRR **abs/2004.08257** (2020)
22. Huang, Y., Chiang, F.: Refining Duplicate Detection for Improved Data Quality. TDDL/MDQual/Futurity@ TPDL (2017)
23. Koumarelas, l., Papenbrock, T., Naumann, F.: MDedup: duplicate detection with matching dependencies. Proceedings of the VLDB Endowment **13**(5), 712–725 (Jan 2020). https://doi.org/10.14778/3377369.3377379, https://dl.acm.org/doi/10.14778/3377369.3377379
24. Krech, D., Grimnes, G.A., Higgins, G., Hees, J., Aucamp, I., Lindström, N., Arndt, N., Sommer, A., Chuc, E., Herman, I., Nelson, A., McCusker, J., Gillespie, T., Kluyver, T., Ludwig, F., Champin, P.A., Watts, M., Holzer, U., Summers, E., Morriss, W., Winston, D., Perttula, D., Kovacevic, F., Chateauneu, R., Solbrig, H., Cogrel, B., Stuart, V.: RDFLib (Aug 2023). https://doi.org/10.5281/zenodo.6845245, https://github.com/RDFLib/rdflib
25. Liseo, B., Tancredi, A.: Some advances on bayesian record linkage and inference for linked data. In: Proceedings of the ESSnet Data Integration Workshop. EUROSTAT (2011)
26. Mel, A., Kang, B., Lijffijt, J., De Bie, T.: FONDUE: A Framework for Node Disambiguation and Deduplication Using Network Embeddings. Applied Sciences **11**(21), 9884 (Jan 2021). https://doi.org/10.3390/app11219884, https://www.mdpi.com/2076-3417/11/21/9884, number: 21 Publisher: Multidisciplinary Digital Publishing Institute
27. Naumann, F., Herschel, M.: An Introduction to Duplicate Detection. Synthesis Lectures on Data Management, Springer International Publishing, Cham (2010). https://doi.org/10.1007/978-3-031-01835-0, https://link.springer.com/10.1007/978-3-031-01835-0
28. Observatorio de valores del suelo para fortalecer la política de Integración social y urbana de barrios populares (Aug 2021), https://www.argentina.gob.ar/noticias/observatorio-de-valores-del-suelo-para-fortalecer-la-politica-de-integracion-social-y

29. Olston, C., Najork, M.: Web Crawling. Foundations and Trends® in Information Retrieval **4**(3), 175–246 (2010). https://doi.org/10.1561/1500000017, http://www.nowpublishers.com/article/Details/INR-017
30. Sadinle, M.: Detecting duplicates in a homicide registry using a Bayesian partitioning approach. The Annals of Applied Statistics **8**(4), 2404–2434 (Dec 2014). https://doi.org/10.1214/14-AOAS779, https://projecteuclid.org/journals/annals-of-applied-statistics/volume-8/issue-4/Detecting-duplicates-in-a-homicide-registry-using-a-Bayesian-partitioning/10.1214/14-AOAS779.full, publisher: Institute of Mathematical Statistics
31. Schrenk, M.: Webbots, Spiders, and Screen Scrapers: A Guide to Developing Internet Agents with PHP/CURL. No Starch Press, second edition edn. (2012)
32. Singhal, A., et al.: Introducing the knowledge graph: things, not strings. Official google blog **5**(16), 3 (2012)
33. Tancredi, A., Liseo, B.: A hierarchical Bayesian approach to record linkage and population size problems. The Annals of Applied Statistics **5**(2B) (Jun 2011). https://doi.org/10.1214/10-AOAS447, https://projecteuclid.org/journals/annals-of-applied-statistics/volume-5/issue-2B/A-hierarchical-Bayesian-approach-to-record-linkage-and-population-size/10.1214/10-AOAS447.full
34. Zhao, B.: Web Scraping. In: Schintler, L.A., McNeely, C.L. (eds.) Encyclopedia of Big Data, pp. 1–3. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-32001-4_483-1, http://link.springer.com/10.1007/978-3-319-32001-4_483-1
35. Zhu, H., Wang, X., Jiang, Y., Fan, H., Du, B., Liu, Q.: FTRLIM: Distributed Instance Matching Framework for Large-Scale Knowledge Graph Fusion. Entropy **23**(5), 602 (May 2021). https://doi.org/10.3390/e23050602, https://www.mdpi.com/1099-4300/23/5/602