



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

SIMILITUD DE ENTIDADES EN WIKIDATA

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS, MENCIÓN COMPUTACIÓN

MARCO ANTONIO CABALLERO GUILLEN

PROFESOR GUÍA:
AIDAN HOGAN

MIEMBROS DE LA COMISIÓN:
BENJAMÍN BUSTOS CÁRDENAS
CLAUDIO GUTIÉRREZ GALLARDO
DIEGO ARROYUELO BILLIARDI

Este trabajo ha sido parcialmente financiado por
Instituto Milenio de Fundamentos de los Datos

SANTIAGO DE CHILE

2021

Resumen

Wikidata es una base de datos de grafos, la cual está formada por entidades (nodos) y relaciones (aristas) que unen las entidades. En el contexto de este tipo de bases de datos, hay varias aplicaciones que dependen de una noción de similitud entre entidades que describen el grafo, por ejemplo, para proveer recomendaciones al usuario. Una medida global de similitud intenta establecer un valor numérico a cada par posible de entidades; dicho valor determina la similitud entre dos entidades. Existen medidas de similitud para grafos con características diferentes a Wikidata; en particular, las medidas existentes no consideran toda la información disponible en Wikidata, como por ejemplo las etiquetas de las aristas que denotan diferentes tipos de relaciones.

En este trabajo se adapta y aplica una medida de similitud bastante utilizada en la actualidad llamada SimRank; se analiza la calidad de sus resultados al ser aplicada en Wikidata, para luego identificar las debilidades y fortalezas de esta medida.

Además de hacer el análisis anterior, se proponen 3 medidas de similitud para ser aplicadas a Wikidata; dichas medidas logran aproximarse en calidad de resultados a SimRank, sin embargo son mejores en rendimiento computacional.

Seguidamente se comparan estas medidas (tres propuestas y SimRank) con dos servicios de recomendación, para intentar analizar si los valores devueltos por las medidas de similitud se aproximan a la percepción de similitud de las personas.

Esta investigación comienza definiendo formalmente las medidas de similitud propuestas, resaltando las fortalezas y debilidades de cada una de ellas. Además se presenta una implementación en Spark para cada una de las medidas; dichas implementaciones fueron utilizadas para realizar los experimentos de las siguientes secciones.

Finalmente se analiza el rendimiento y calidad de los resultados de cada una de las medidas propuestas, para luego continuar a comparar estas medidas con SimRank, y poder concluir cuáles son los aspectos en que las medidas propuestas superan esta medida base.

Este trabajo está dedicado a:

A Dios, porque su palabra dice: "Y todo lo que hagáis, hacedlo de corazón, como para el Señor y no para los hombres". Y yo agrego con excelencia para Dios y para los hombres.

A mi esposa Sayda, por apoyarme en esta aventura, por acompañarme siempre a la distancia, y por ser madre y padre de nuestra hija mientras yo realizaba mis estudios.

A mi hija Valery por esperarme pacientemente en sus 2 primeros años de vida, tiempo que no recuperaremos, pero esperando que valga la pena en un futuro.

A mi madre por su apoyo incondicional frente a cualquier situación, por tener fe en mí, y por ser siempre el salvavidas de mi vida.

Agradecimientos

A Dios, por darme vida, salud y la oportunidad de estar en este lugar para estudiar e investigar. Sin su bendición hubiera sido imposible estar aquí y concluir mi trabajo.

Al sistema becas Internacionales 20/20, que financió gran parte de los costos de mi estadía en Chile.

A cada uno de los alumnos, docentes y funcionarios del DCC de la Universidad de Chile, por acogerme y apoyarme en cada uno de los espacios que compartimos.

A mi tutor Aidan Hogan, por su apoyo sincero, por exigirme siempre la excelencia con una amabilidad admirable, y por darme la oportunidad de investigar a su lado.

Tabla de Contenido

Introducción	1
1. Objetivos e Hipótesis	4
1.1. Objetivo General	4
1.2. Objetivos Específicos	4
1.3. Hipótesis	5
1.4. Preguntas de Investigación	5
1.5. Metodología	5
2. Preliminares	7
2.1. La Web Semántica	7
2.2. RDF	7
2.2.1. Vocabulario RDF	8
2.2.2. Esquema RDF	9
2.3. Wikidata	9
2.4. Apache Spark	11
3. Trabajos Relacionados	13
3.1. Medidas Globales de Similitud de Vértices	13
3.2. Comparaciones de Medidas de Similitud	16
3.3. Aplicaciones de medidas de similitud	17
3.4. Investigaciones en Wikidata	18
3.5. Aporte de esta Investigación	19
4. Algoritmos de Similitud Propuestos	20
4.1. Preliminares	20
4.2. Conteo de Vecinos Comunes	21
4.3. Suma Probabilística de Vecinos Comunes	22
4.4. Conteo de Intersección	24
4.5. SimRank	25
4.6. Comparaciones de las Medidas Propuestas	26
5. Implementación de las Medidas	27
5.1. Extractor de Tuplas	27
5.2. Conteo de Vecinos Comunes	28
5.3. Suma Probabilística de Vecinos Comunes	29
5.4. Conteo de Intersección	29

5.5. Costo de las Implementaciones	31
6. Evaluación	33
6.1. Descripción de Datos	33
6.2. Tiempos en Calcular las Medidas	34
6.3. Efecto del valor de k en las medidas.	38
6.3.1. Coeficiente de Correlación tau-b de Kendall	38
6.3.2. Valor de p y la significación estadística	39
6.3.3. Resultados de las Pruebas	40
6.3.4. Análisis del efecto de k	42
6.4. Análisis de la Calidad de los Resultados	43
6.5. Comparación de los Resultados con Medida Base	46
6.6. Aplicación de las medidas en aristas inversas	49
Conclusión	54
Bibliografía	56

Índice de Tablas

2.1. Ejemplos de IRI. Basado en [10]	8
6.1. Descripción de Dominios Utilizados.	33
6.2. Tiempo en segundos para Películas	34
6.3. Tiempo en segundos para Universidades	35
6.4. Tiempo en segundo para Países	36
6.5. Tiempo en segundos para Álbumes de Música	37
6.6. Efecto de k en Conteo de Vecinos Comunes	40
6.7. Efecto de k en Suma Probabilística	41
6.8. Efecto de k en Conteo de Intersección	42
6.9. Datos para prueba de Calidad	44
6.10. Ejemplo de Datos	44
6.11. Resultados de Correlación en BestSimilar	45
6.12. Resultados de Correlación en Last.fm	45
6.13. Descripción de Datos utilizado para SimRank.	46
6.14. Comparación de las medidas propuesta con SimRank.	47
6.15. Resultados de Calidad con Datos Reducidos en Películas	47
6.16. Resultados de Correlación con Datos Reducidos en Álbumes de Música	48
6.17. Tiempo en segundos de Pruebas de Correlación con Datos Reducidos	48
6.18. Descripción de dominios con arista en dos direcciones.	50
6.19. Descripción de Datos utilizado para SimRank con aristas en dos direcciones.	50
6.20. Correlación de Resultados con Aristas en dos Direcciones.	50
6.21. Tiempo en Segundos para Películas con Aristas en dos Direcciones	51
6.22. Tiempo en Segundos para Álbumes de Música con Aristas en dos Direcciones	52
6.23. Resultados de Correlación con Datos Reducidos en Películas con Aristas en dos Direcciones	52
6.24. Resultados de Correlación con Datos Reducidos en Álbumes de Música con Aristas en dos Direcciones	53

Índice de Ilustraciones

1.	Ejemplo de tres películas donde actúan Clint Eastwood y Morgan Freeman. .	2
1.1.	Pasos a seguir	5
2.1.	Muestra de la información de Golpes del Destino	10
3.1.	Fase de Construcción de SimRank; ejemplo tomado de [17]	14
3.2.	Fase de Asignación de Valor de Similitud de SimRank; ejemplo tomado de [17]	15
3.3.	Ejemplo de la medida de Leicht et al. [19]	15
4.1.	Nueva representación de la información	21
4.2.	Vecindad de Los Imperdonables	22
4.3.	Ejemplo de Dependencia de (p, o)	24
6.1.	Tiempo en Películas	35
6.2.	Tiempo en segundos para Universidades	36
6.3.	Tiempo en segundo para Países	37
6.4.	Tiempo en segundos para Álbumes de Música	38
6.5.	Efecto de k en vecinos comunes	40
6.6.	Efecto de k en Suma Probabilística	41
6.7.	Efecto de k en Conteo de Intersección	42
6.8.	Ejemplo de Recomendación BestSimilar	43
6.9.	Ejemplo de Recomendación Last.fm	43
6.10.	Resultados de Correlación en BestSimilar	45
6.11.	Resultados de Correlación en Last.fm	46
6.12.	Resultados de Correlación con Datos Reducidos en BestSimilar	47
6.13.	Resultados de Correlación con Datos Reducidos en Last.Fm	48
6.14.	Tiempo en segundos de Pruebas con Datos Reducidos	49
6.15.	Correlación de Resultados con Aristas en dos Direcciones.	51
6.16.	Resultados de Calidad en BestSimiliar con Aristas en Dos Direcciones. . . .	52
6.17.	Resultados de Correlación en Last.fm con Aristas en dos Direcciones.	53

Introducción

La Web ha sido un lugar donde han nacido varias iniciativas importantes. Entre ellas, Wikipedia ¹, una enciclopedia en línea gratuita que cualquiera en todo el mundo puede editar [4], es uno de los cinco principales sitios en la Web ². Wikipedia permite a los usuarios, llamados colaboradores, crear, editar y publicar documentos. Esos documentos pueden ser leídos por cualquier persona en la Web. Wikipedia se creó en 2001. Desde entonces, se han publicado más de 47 millones de artículos en más de 290 idiomas ³.

Berners-Lee no solo participó en la creación de la Web sino que también articuló otra visión: la Web Semántica. Mientras que la Web es una red de documentos diseñados para ser leídos por humanos, la Web Semántica es una red de datos diseñada para ser leída y manipulada tanto por humanos como por computadoras [6]. En el sueño de Berners Lee, un agente, un software especializado, podía realizar tareas, recuperar información, programar citas, comprender datos, realizar razonamientos automatizados, etc.

La Web Semántica utiliza bases de datos de grafos, específicamente basadas en RDF [10] para almacenar información. A diferencia de las bases de datos relacionales, donde la fijación del esquema es una de las primeras tareas, las bases de datos de grafos no necesitan establecer ningún esquema, por lo que su flexibilidad es una de sus características principales.

Sin embargo, cada motor de base de datos relacional tiene comandos para enumerar todas las tablas y mostrar los detalles de una en específico, es decir, es una tarea fácil ver el esquema y un resumen de los datos. Además, cuando se agrega un nuevo registro a una base de datos relacional, debe ajustarse al esquema fijo y satisfacer las condiciones de integridad de los datos (registrados en el mismo esquema).

Esto no ocurre en las bases de datos de grafos, donde generalmente no hay un esquema fijo ni restricciones de datos. Debido a esta flexibilidad, surgen algunos problemas: las bases de datos de grafos son difíciles de consultar, comprender, explorar y resumir. No es fácil ver qué contiene una base de datos de grafos, cómo evolucionará o si sus datos están completos.

Wikidata es una base de datos de grafos, la cual está formada por entidades (nodos) y relaciones (aristas) que unen las entidades. [34]. Wikidata contiene más de 45 millones de elementos de datos [5]. Actúa como el centro de interconexión de las páginas de Wikipedia sobre un elemento específico en diferentes idiomas. Automatiza funciones como las infoboxes

¹<https://www.wikipedia.org/>

²<https://www.alexa.com/topsites>. Consultado en 22.09.2020.

³https://meta.wikimedia.org/wiki/List_of_Wikipedias. consultado en 22.09.2020.

en Wikipedia y se utiliza cada vez más para otras aplicaciones, como el enriquecimiento de datos y la respuesta a preguntas.

Al buscar en Wikidata nos encontramos con entidades que al parecer son similares, ¿Pero son similares? O ¿Solo es mi percepción de las entidades?; es aquí donde nos surge la necesidad de una escala uniforme para medir la similitud de dos entidades, y de esta manera poder decir “Con base en la escala x estas dos entidades son muy similares”, y ver si existe un consenso entre los usuarios sobre los resultados de esta medida de similitud.

Las métricas de similitud son utilizadas para apoyar los algoritmos recomendadores; de igual manera se utilizan para eliminar entidades duplicadas o reducir redundancia de datos.

Ahora, por la naturaleza de Wikidata, se podría estimar la similitud entre pares de nodos de diferentes maneras; sin embargo, este trabajo está interesado en el concepto de similitud que propone Leicht et al en [19], el cual establece que dos vértices son similares si sus vecinos inmediatos en la red son similares; la cual se describe más a detalle en capítulos siguientes.

En las siguientes imágenes se muestra parte de la información sobre tres películas donde actúan Clint Eastwood y Morgan Freeman.

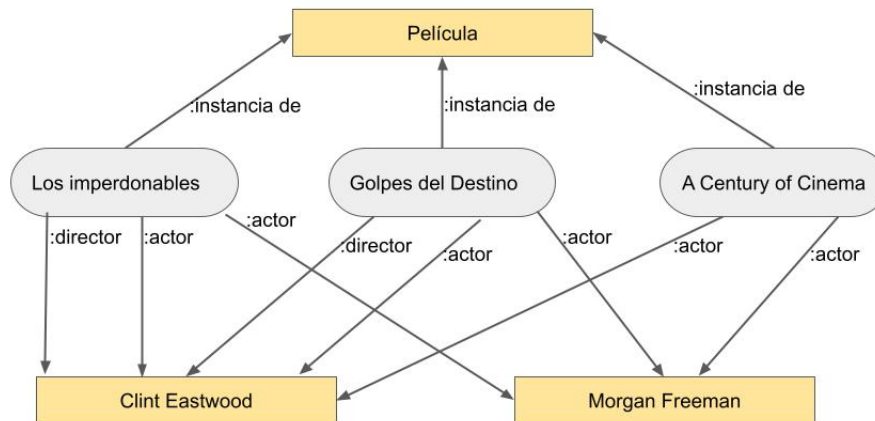


Figura 1: Ejemplo de tres películas donde actúan Clint Eastwood y Morgan Freeman.

Una forma de medir la similitud entre entidades es contar lo que tienen en común, pero ¿Cómo hacemos esto en Wikidata?

Ignorando la información de las aristas, si contamos el número de vecinos que los pares de nodos tiene en común, vemos que las 3 películas de la Figura 1, tienen la misma similitud, ya que todas comparten los vecinos $\{Clint\ Eastwood, Morgan\ Freeman, Películas\}$. Sin embargo **Los Imperdonables** y **Golpes Del Destino** tienen a Clint Eastwood tanto de actor como director.

Por lo anterior surgen las preguntas ¿Qué se debe considerar para medir la similitud en Wikidata? ¿Cómo se comportan las medidas existentes si las aplicamos en Wikidata?

Otro detalle a considerar es ¿Qué tan lógicos son los resultados de los algoritmos de similitud? ¿Cómo se compara los resultados con sistemas recomendadores que toman en cuenta las opiniones de los usuarios?

Capítulo 1

Objetivos e Hipótesis

Los índices de similitud en bases de datos de grafos se pueden clasificar de diferentes maneras [20], sin embargo podemos diferenciar dos grupos. Primero están los algoritmos que se basan en los nodos vecinos, es decir son algoritmos que se mueven a través del grafo utilizando las aristas, pero ignoran qué representa esta arista. Segundo están los algoritmos que utilizan la información de la arista; no solo se mueven a través de ella como un simple camino, sino que usan la información que la arista provee.

Ahora por la naturaleza de Wikidata las aristas representan relaciones diferentes, que son importantes de considerar, por lo que las siguientes preguntas son importantes: ¿Se pueden proponer medidas de similitud que usen la información de las aristas en Wikidata? ¿Se puede aplicar o adaptar los algoritmos utilizados en grafos simples (no dirigidos y sin información en la arista) a Wikidata?

Algo importante a considerar al manejar un algoritmo es su escalabilidad, más aún trabajando en una base de datos tan grande como Wikidata, por lo que también en este estudio se pretende hacer énfasis en este aspecto, buscando algoritmos que proveen una buena medición de la similitud pero además son escalables.

1.1. Objetivo General

- Proponer, aplicar y comparar medidas de similitud de entidades en Wikidata, basadas en la noción de similitud de vértices de Leicht et al. [19].

1.2. Objetivos Específicos

- Identificar medidas existentes de similitud entre vértices de un grafo, que pueden ser aplicadas en Wikidata.
- Aplicar y evaluar esas medidas existentes sobre Wikidata para entender sus fortalezas y debilidades.

- Proponer nuevas medidas que mejoren las debilidades encontradas con las medidas existentes.
- Comparar los resultados de las medidas respecto a su escalabilidad.
- Comparar los resultados de las medidas respecto a las similitudes que encuentran.

1.3. Hipótesis

- Es posible adaptar medidas de similitud de vértices utilizadas en otro tipo de grafos a Wikidata.
- Es posible proponer nuevas medidas de similitud que mejoren los resultados de las medidas existentes.

1.4. Preguntas de Investigación

- ¿Es posible adaptar medidas utilizados en otro tipo de grafos a Wikidata?
- ¿Son las medidas que utilizan la información de las aristas mejores que las medidas que solo se basan en la información de los nodos para medir la similitud en Wikidata?

1.5. Metodología

Las actividades a realizar aparecen en la Figura 1.1.

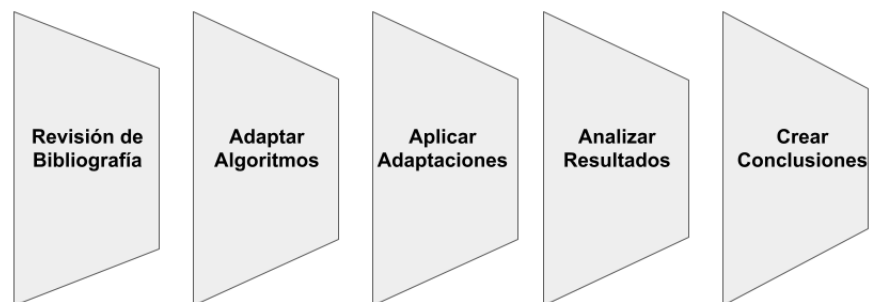


Figura 1.1: Pasos a seguir

1. **Revisión Bibliografía:** Se comenzará buscando métricas de similitud aplicadas en otros grafos, considerando su escalabilidad y su posible aplicación a Wikidata.
2. **Adaptar Algoritmos:** En este punto se trabajará con datos de Wikidata, pero en dominios específicos, para así poder realizar las adaptaciones de los algoritmos de forma más fácil y rápida, y de esta manera asegurarse que los algoritmos funcionen correctamente.
3. **Aplicar Adaptaciones:** Aquí se realizarán mediciones de similitud de entidades con una cantidad de datos mayor; no se puede asegurar que se aplicarán las adaptaciones sobre la totalidad de los datos, ya que se puede encontrar con algoritmos cuyo costo es alto, pero la métrica tiene un importante aporte a nuestro estudio.
4. **Análisis de Resultados:** Se procederá a comparar los resultados de la etapa anterior con respecto al rendimiento y similitud que encuentran los algoritmos.
5. **Conclusiones:** Se crearán conclusiones que intenten responder a las preguntas de investigación.

Capítulo 2

Preliminares

Este capítulo tiene como objetivo describir algunos conocimientos básicos de la Web Semántica y RDF, para ayudar a entender algunos términos que se utilizarán en capítulos siguientes, aclarando que sólo se describen conocimientos básicos necesarios para este estudio.

2.1. La Web Semántica

Como se mencionó anteriormente la Web Semántica es una red de datos interconectados diseñados para ser leídos y manipulados tanto por humanos como por computadoras. En palabras de Berners-Lee: "La Web Semántica es una extensión de la web actual en la que la información tiene un significado bien definido, que permite que las computadoras y las personas trabajen mejor en cooperación"[6].

Para lograr esto en la Web Semántica, los datos se estructuran de manera tal que permite a los agentes de software recuperar, analizar, interpretar, manipular y comunicar datos, con el objetivo de realizar tareas sofisticadas de manera automatizada [28]. En otras palabras, si la visión de la Web Semántica se realizara plenamente, podríamos hacer consultas complejas y recibir respuestas directas basadas en datos integrados automáticamente desde múltiples fuentes en la Web.

2.2. RDF

El Marco de Descripción de Recursos (RDF) es el marco utilizado para representar información sobre recursos, también llamados entidades en la Web Semántica. Según Decker et al. [11] su motivación es proporcionar una representación estándar e invariante de las entidades en la Web Semántica, para facilitar la interpretación o manejo de la información por parte de las computadoras.

La construcción básica en RDF es un triple de (sujeto-predicado-objeto): un sujeto s tiene un atributo p con valor o . Tal triple corresponde a la relación que comúnmente se escribe como (s, p, o) . Una tercera forma de pensar sobre un triple RDF tan básico es como una arista etiquetada entre dos nodos en un grafo: $s \xrightarrow{p} o$. Esta última notación es particularmente útil, ya que RDF permite mezclar sujetos y objetos (primero y tercer elemento de los triples RDF básicos): cualquier objeto puede desempeñar el papel de un sujeto.

Para ejemplificar, las triples que describen la información presentada en la Figura 1.1 son:

```

1 Los Imperdonables , instancia de , Pelicula .
2 Golpes del Destino , instancia de , Pelicula .
3 A Century of Cinema , instancia de , Pelicula .
4 Los Imperdonables , actor , Clint Eastwood .
5 Los Imperdonables , director , Clint Eastwood .
6 Los Imperdonables , actor , Morgan Freeman .
7 Golpes del Destino , actor , Clint Eastwood .
8 Golpes del Destino director , Clint Eastwood .
9 Golpes del Destino , actor , Morgan Freeman .
10 A Century of Cinema , actor , Clint Eastwood .
11 A Century of Cinema , actor , Morgan Freeman .

```

2.2.1. Vocabulario RDF

Un vocabulario RDF es una colección de IRIs (Identificador Internacional de Recursos) destinados a ser utilizados en grafos RDF [10]. A menudo comienzan con una subcadena. Algunos IRIs están asociados con un prefijo. La Tabla 2.1 muestra algunos ejemplos de vocabularios RDF. Estos prefijos se usan comúnmente para abreviar IRI. Por ejemplo, <http://www.wikidata.org/wiki/Q172241> generalmente se abrevia como `wd: Q172241`.

Prefijo	IRI	Vocabulario RDF
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	El vocabulario RDF incorporado
rdfs	http://www.w3.org/2000/01/rdf-schema#	El vocabulario del esquema RDF
xsd	http://www.w3.org/2001/XMLSchema#	Los tipos XSD compatibles con RDF
foaf	http://xmlns.com/foaf/0.1/	Vocabulario de amigo de un amigo
wd	http://www.wikidata.org/wiki/	Entidades en Wikidata.
owl	http://www.w3.org/2002/07/owl#	Lenguaje de ontología web

Tabla 2.1: Ejemplos de IRI. Basado en [10]

2.2.2. Esquema RDF

El Esquema de RDF (RDFS) es una extensión semántica de RDF. Proporciona mecanismos para describir grupos de recursos relacionados, y las relaciones entre estos recursos [7]. Mientras que RDF define `rdf:type` y `rdf:Property` entre otros términos, RDFS define `rdfs:Resource`, `rdfs:Class`, `rdfs:Literal`, `rdfs:subClassOf`, `rdfs:subPropertyOf`, `rdfs:domain`, `rdfs:range`, etc.

Todas las cosas descritas por RDF son instancias de la clase `rdfs:Resource`. Una clase es un conjunto de recursos que comparten algunos puntos en común; un ejemplo podría ser países, películas, actores, etc. Los recursos que son clases se expresan con `rdfs:Class`, mientras que `rdf:Property` es la clase de todas las propiedades, que son relaciones binarias entre un sujeto y un objeto. Las propiedades `rdfs:subClassOf` y `rdfs:subPropertyOf` se utilizan para expresar jerarquías entre clases y propiedades. Finalmente, `rdfs:domain` y `rdfs:range` se usan para expresar las clases de sujetos y las clases de objetos relacionados por una propiedad.

A modo de ejemplo, se presentan algunas triples de RDFS que se podrían aplicar a la información presentada en de la Figura 1.1

```
1 Pelicula , rdfs:subClassOf , Show .
2 Pelicula , rdf:type , rdfs:Class
3 Show , rdf:type , rdfs:Class
4 director , rdfs:subPropertyOf , cast
5 director , rdfs:domain , Show
6 director , rdfs:range , Persona
7 director , rdf:type , rdf:Property
```

2.3. Wikidata

Wikipedia permite a los usuarios, llamados colaboradores, crear, editar y publicar documentos. Una vez que se publican estos documentos, cualquiera puede leerlos en la Web. Wikipedia se creó en 2001. Desde entonces, se han publicado más de 47 millones de artículos en más de 290 idiomas ¹.

Al estar basado en la Web (de Documentos), Wikipedia fue diseñada para ser leída por humanos. En consecuencia, su contenido es apenas interpretable por máquinas. Además, se para diferentes idiomas, proporcionando una Wikipedia para cada dialecto. Específicamente, hay una Wikipedia en español ², otra en inglés ³, otra en alemán ⁴, y así sucesivamente.

Para resolver estos y otros problemas, se creó una Wikipedia semántica, llamada Wikidata [15]. Wikidata puede verse como una Wikipedia para datos, que gestiona información en diferentes idiomas sin perder la edición abierta y el control comunitario que son las carac-

¹https://meta.wikimedia.org/wiki/List_of_Wikipedias. Consultado en 12.01.2020

²<https://es.wikipedia.org>

³<https://en.wikipedia.org>

⁴<https://de.wikipedia.org>

terísticas distintivas de Wikipedia. Hoy, Wikidata tiene más de 45 millones de páginas, tal que cada página describe una entidad (Clint Eastwood, Los Imperdonables, etc) , y ha sido editado por más de 35 mil editores ⁵.

La Figura 2.1 es una parte de la información en Wikidata de la película Golpes de Destino, donde el código *Q184255* es el identificador de esta entidad, y el predicado *instanceOf* es quien determina la clase a la que pertenece la entidad .

Wikidata proporciona dumps de datos grandes y heterogéneos semanalmente⁶. Mientras Wikidata pone a disposición diferentes tipos de datos y en varios formatos, para los experimentos presentados en esta tesis, se usó el archivo con fecha 20/11/2019, en formato *truthy* dumps de N-Triples [14]. En el formato *truthy*, se mantiene sólo la información más actualizada de Wikidata. Por ejemplo, aunque en Wikidata mantiene datos históricos sobre poblaciones de ciudades, el formato *truthy* sólo provee las poblaciones más recientes.

Million Dollar Baby (Q184255)

2004 film by Clint Eastwood edit

[In more languages](#)
Configure

Language	Label	Description	Also known as
English	Million Dollar Baby	2004 film by Clint Eastwood	
Spanish	Million Dollar Baby	película de 2004 dirigida por Clint Eastwood	Golpes del Destino

[All entered languages](#)

Statements

instance of film edit

[1 reference](#)

[+ add value](#)

title Million Dollar Baby (English) edit

[0 references](#)

[+ add reference](#)

[+ add value](#)

Figura 2.1: Muestra de la información de Golpes del Destino

⁵<https://stats.wikimedia.org/v2/#/wikidata.org>. Accessed on 13.02.2020

⁶https://www.wikidata.org/wiki/Wikidata:Database_download. Accessed on 13.03.2020

2.4. Apache Spark

Debido al tamaño del dump trabajado, y los objetivos de este estudio, se trabajó con Apache Spark para realizar los experimentos; a continuación se describen generalidades de este framework.

Apache Spark [37] es un motor de clúster de propósito general que es muy rápido y confiable. Este sistema proporciona interfaces en varios lenguajes de programación como Java, Python, Scala. Spark está especializada en agilizar el análisis de datos.

Inicialmente, el sistema se desarrolló en UC Berkeley como proyecto de investigación y rápidamente adquirió el estado de incubación en Apache en junio de 2013 [29].

Las funciones del motor Spark son bastante avanzadas y diferentes a las de Hadoop. El motor Spark está desarrollado para el procesamiento en memoria, así como un procesamiento basado en disco. Esta capacidad de procesamiento en memoria lo hace mucho más rápido que cualquier motor de procesamiento distribuido de datos tradicional (como Hadoop).

El conjunto de datos distribuidos (RDD) es la estructura de datos paralela y tolerante a fallas de Spark [36]. Spark distribuye automáticamente los datos en el cluster RDD y realiza operaciones paralelas en ellos. Los RDDs pueden contener cualquier objeto o clase de Python, Java o Scala. Una de las capacidades más importantes de Spark es el almacenamiento en caché de un conjunto de datos en la memoria principal ⁷. Los RDDs almacenados en caché son tolerantes a fallos y los nodos del clúster los procesan mucho más rápido que los datos en el disco duro.

RDD admite dos tipos de operaciones: transformaciones que generan un nuevo conjunto de datos a partir de uno existente, y acciones que devuelven un valor de un conjunto de datos ⁸. Por ejemplo, el **map** es una transformación que pasa cada elemento de un RDD a una función y da como resultado un nuevo RDD con los valores calculados. Por el contrario, **reduce** es una acción que pasa cada elemento de un RDD a una función y, como resultado, devuelve un solo valor.

Enmarcadas en la clasificación anterior, a continuación se presenta un resumen de las funciones utilizadas en la implementación de las medidas propuestas en este trabajo.

1. **map**($\langle a, b \rangle$) : Dada un Función $F(\langle a, b \rangle) \rightarrow \langle c, d \rangle$, por cada par en el primer map, devuelve un nuevo par según F , dando como resultado un nuevo map.
2. **swap**($\langle a, b \rangle$) : Devuelve el inverso de un par: dado el par $\langle a, b \rangle$ devuelve $\langle b, a \rangle$.
3. **groupByKey**($\langle a, b \rangle$) : Agrupa los pares que tiene la misma llave, devolviendo pares que tiene una llave y como valor tiene un arreglo. Por ejemplo, si el mapa al que se le aplica esta función es $\{(a, b), (a, c), (b, c)\}$ devolvería el siguiente mapa $\{(a, [b, c]), (b, [c])\}$
4. **filter**($\langle a, b \rangle$) : Dado un mapa $\langle a, b \rangle$ y una condición, esta función filtra los pares de ese mapa que cumplan la condición.
5. **flatMapToPair**($\langle a, b \rangle$) : Es similar al map, con la diferencia de que por cada par del

⁷ <https://databricks.com/spark>

⁸ <http://spark.apache.org/docs/latest/programming-guide.html>

mapa $\langle a, b \rangle$, la función de conversión puede devolver más de un resultado, devolviendo un nuevo mapa con cero o más pares para cada par en la entrada.

También proporciona una gran cantidad de herramientas impresionantes de alto nivel [29], como la herramienta de aprendizaje automático MLib, el procesamiento de datos estructurados, SparkSQL, el procesamiento de grafos con GraphX, el motor de procesamiento de flujo llamado Spark Streaming, etc.

Capítulo 3

Trabajos Relacionados

En este capítulo hablaremos de trabajos relacionados sobre medidas globales de similitud de vértices, formas de comparar estas medidas, y sus aplicaciones.

3.1. Medidas Globales de Similitud de Vértices

En esta sección se encuentran descritas algunas medidas de similitud de vértices; cada una de ellas calcula globalmente la similitud, es decir calculan la similitud de cada par de nodos en el grafo.

SimRank [17] es una medida de similitud, aplicable en cualquier dominio con relaciones de objeto a objeto, que mide la similitud del contexto estructural en el que se producen los objetos, en función de sus relaciones con otros objetos; es decir, es una medida que dice: dos objetos son similares si están relacionados con objetos similares. SimRank se utilizó en este estudio, se describe a detalle en la sección 4.5, sin embargo en las siguientes figuras se explicará generalmente esta medida.

En la Figura 3.1 a), se observa un pequeño ejemplo de un grafo G no etiquetado, que representa la relación entre dos profesores (PA, PB) con dos estudiantes (SA, SB) que estudian en una Universidad (U).

Inicialmente SimRank calcula G^2 , el cual está conformado por nodos que representan pares de nodos de G , por ejemplo: (U, U) , (U, PA) y (SA, SB) ; luego se traza un arista entre dos pares (a, b) , (c, d) si se cumple que $a \rightarrow c$ y $b \rightarrow d$ en G ; es decir para trazar una arista entre dos pares, la primera componente de un par debe estar relacionada con la primera componente del segundo par en G ; además se debe cumplir la misma propiedad con la segunda componente de cada par. En la Figura 3.1 b) se muestra el grafo G' , el cual es el grafo que cumple con la construcción antes descrita. No se utilizan todos los posibles pares de G^2 , ya que se excluyeron los pares que no tiene conexión con otro par, por que se necesita trabajar con un grafo conexo G' en la siguiente fase de SimRank.

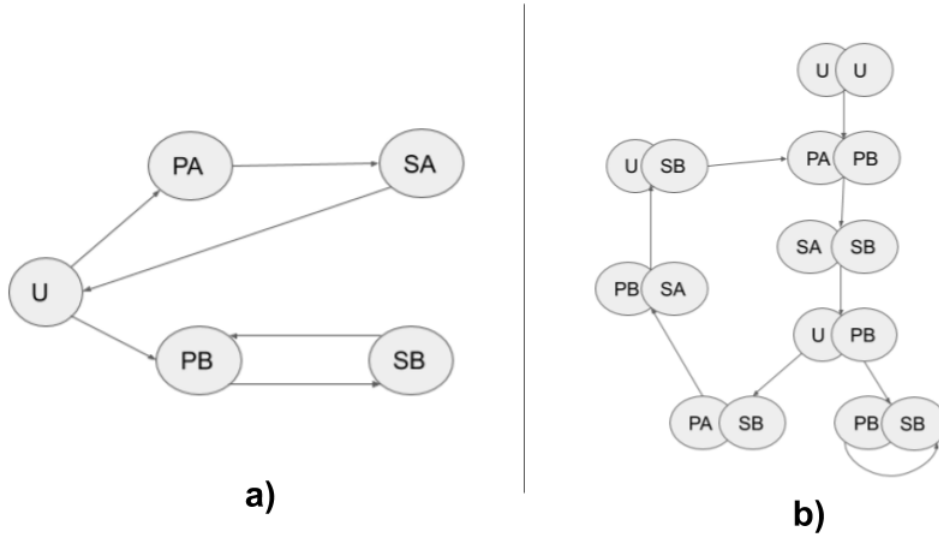


Figura 3.1: Fase de Construcción de SimRank; ejemplo tomado de [17]

Luego SimRank comienza a asignar valores de similitud a cada nodo de G' ; se inicia con el caso base ($k = 0$) donde asignamos una similitud de 1 si son pares de la forma (a, a) , y al resto de pares una similitud de 0. Según Jeh y Widom [17] esto se debe a que la similitud de una entidad consigo misma es 1; esto se puede ver en la Figura 3.2 a). Luego en el siguiente paso ($k = 1$), tomando los valores asignados en el paso anterior, se le asigna como valor de similitud a cada nodo de G' , el promedio de similitud de los vecinos que llegan a ese nodo, multiplicado por una constante C (en el paper original, se define $C = 0,8$) de decaimiento de transferencia. Por ejemplo, cuando $k = 1$ la similitud de $(PA, PB) = \frac{1+0}{2} \times 0,8 = 0,40$, que es igual al promedio de las similitudes de los pares que llegan a ese nodo, multiplicado por una constante de transferencia. Siguiendo con esta idea, se puede llegar a asignar un valor de similitud a cada nodo de G' , lo que se muestra en la Figura 3.2 c).

Después de $k = 6$, SimRank puede seguir actualizando las medidas de similitud de los nodos esperando que en algún momento los valores converjan.

Se puede ver que el costo computacional de una implementación directa de SimRank en el caso peor es de $O(n^2)$ en memoria donde n es el número de nodos de G , ya que en el caso de un grafo completo se debe almacenar un valor de similitud para cada par de nodos de G^2 ; en tiempo esta medida tiene un costo de $O(kn^4)$ donde k es el número de iteraciones, ya que si tenemos un grafo completo, es decir, uno en el que todos sus vértices se relacionen entre sí, entonces para calcular la medida de cada nodo de G^2 tendremos que sumar las similitudes de todos los nodos de G^2 en cada iteración; además una desventaja de esta medida es la construcción inicial de G' , ya que aunque no siempre usemos todos los nodos de G^2 en el caso general se tiene que trabajar con una matriz de adyacencia de tamaño n^2 .

Otro método interesante es usar la noción de *similitud estructural* [19], el cual propone una medida de similitud basada en el concepto de que dos vértices son similares si sus vecinos

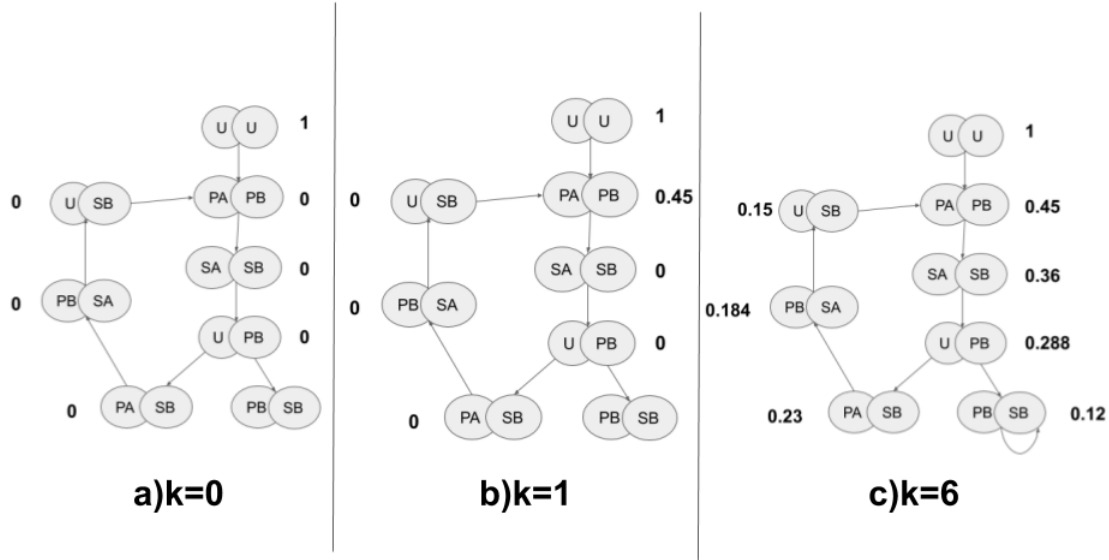


Figura 3.2: Fase de Asignación de Valor de Similitud de SimRank; ejemplo tomado de [17]

inmediatos en la red son similares; la idea de esta medida se ejemplifica en la figura 3.3, donde un vértice j es similar al vértice i (línea discontinua) si tiene un vecino de red v (línea continua) que es similar a j . Además Leicht et al. [19] nos muestra la aplicación de este método a ciertos problemas, como por ejemplo encontrar sinónimos en un diccionario basándose en su definición, concluyendo que los resultados parecen indicar que la medida es capaz de extraer información útil sobre la similitud de los vértices en función de la topología de la red.

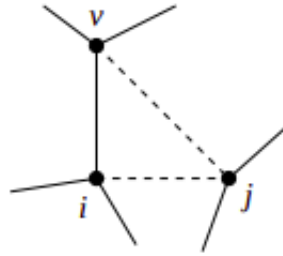


Figura 3.3: Ejemplo de la medida de Leicht et al. [19]

Petrova et al. [26] proponen un marco formal para la comparación de entidades independientes del dominio sobre los grafos RDF. Modelan similitudes y diferencias entre entidades como consultas SPARQL que satisfacen ciertas propiedades adicionales, y proponen algoritmos para calcularlas.

En cuanto a la estructura del grafo, Chebotarev y Shamis [9] proponen la medida de similitud Matrix Forest Index, que establece que la similitud entre (A, B) dos nodos del grafo, es igual a la cantidad de árboles con raíz en A que tenga a B .

Luego tenemos la medida Random Walk with Restart (RWR) [33]; esta medida es una aplicación directa del algoritmo PageRank [13] que considera un caminante aleatorio y define la similitud entre dos nodos (A, B) como la suma entre la probabilidad de que el caminante llegue de A a B más la probabilidad de que llegue de B a A .

Si bien en esta sección no se muestran todas las medidas de similitud para grafos existentes, se muestran diferentes aproximaciones a esta medida, ya sea usando el grafo, consultas o caminatas aleatorias; sirvió para dar una base clara de las posibles variantes que se pueden utilizar para proponer una medida de similitud. Cabe destacar que la mayoría de los métodos han sido propuestos en el contexto de grafos simples o grafos dirigidos, y no toman en cuenta las etiquetas de arcos disponibles en el caso de modelos como RDF.

3.2. Comparaciones de Medidas de Similitud

A continuación se muestran investigaciones que comparan medidas de similitud; estos trabajos nos dan una idea de cómo comparar medidas de similitud. ¿Qué metodología usar? ¿Qué elementos considerar? Estas preguntas son necesarias de responder para nuestro estudio, ya que además de proponer medidas de similitud, buscamos comparar esta medida propuesta con las medidas existentes.

Existe un estudio comparativo [20] donde se encuentran diferentes algoritmos de similitud aplicados en grafos; cabe recalcar que estos algoritmos son aplicados a grafos no dirigidos, sin embargo fue un buen punto de partida para comenzar nuestra investigación, ya que en este documento se encuentra un buen banco de algoritmos de similitud y algunas aplicaciones de estos.

Milo et al. [35] revisan SimRank, partiendo de que se ha observado que, para muchas aplicaciones SimRank puede generar estimaciones de similitud inexactas, debido al hecho de que se enfoca en la estructura de la red e ignora la semántica transmitida en las etiquetas de nodo / borde. Por lo tanto, la pregunta que se hace es ¿SimRank puede enriquecerse con la semántica y preservar sus ventajas? Respondiendo a esta interrogante se propone SemSim, una variante modular de SimRank que permite inyectar en el cálculo cualquier medida semántica similar.

Intentando calcular eficientemente SimRank, se han realizado varios intentos durante la última década de refinar o aproximar esta medida. Esto ha llevado a muchos algoritmos propuesto por investigadores que calculan o aproximan eficientemente SimRank. Zhang et al. [38] realizan un estudio para comparar estos algoritmos para comprender sus ventajas y desventajas, implementando diez algoritmos publicados de 2002 a 2015, y comparándolos usando grafos sintéticos y del mundo real. Muchas de las optimizaciones propuestas para SimRank en la literatura funcionan en el caso de tener un nodo de entrada y buscar los nodos más similares (fuente única), mientras que estamos interesados en la medida global, es decir, calcular la similitud de todos los pares de nodos sin elegir un nodo de entrada.

3.3. Aplicaciones de medidas de similitud

Las aplicaciones de las medidas de similitud han sido varias y han dado diferentes resultados; en esta sección se presentan ciertas investigaciones que tiene como objetivo usar medidas de similitud para resolver problemas de alta complejidad.

Skopal y Bustos [32] presentan un artículo donde analizan el empleo de funciones de similitud no métricas para la búsqueda de similitudes efectiva y eficiente en dominios complejos, concluyendo que la medición de similitud no métrica se usa ampliamente en dominios aislados, que abarcan muchas áreas de investigación interdisciplinaria. Esto incluye bases de datos multimedia, series temporales, tareas médicas, científicas, químicas y bioinformáticas, entre otras.

Ejemplo de estas aplicaciones tenemos que SimRank fue utilizado por Fogaras y Rácz [12] para explotar la información de similitud oculta en la estructura de enlaces de la Web. La similitud de los vecindarios de vértices de varios pasos se evalúa numéricamente mediante funciones de similitud que incluyen SimRank; además se incluye una nueva variante con mejores características teóricas; y el coeficiente de Jaccard, extendido a vecindarios de pasos múltiples. Los resultados experimentales sugieren que la estructura de hipervínculo de los vértices dentro de cuatro a cinco pasos proporciona información más adecuada para la búsqueda de similitud que los vecindarios de un solo paso.

Relacionado a la anterior, Sinha y Mihalcea [31] presentan evaluaciones comparativas usando varias medidas de la similitud semántica de las palabras y varios algoritmos para la centralidad de los grafos. Los resultados indican que la combinación correcta de medidas de similitud y algoritmos de centralidad de grafos, pueden conducir a un rendimiento que compita con el estado del arte en la desambiguación del sentido de la palabra no supervisada, medida en conjuntos de datos estándar.

Se han abordado dos problemas fuertemente interdependientes con medidas de similitud: la relación semántica y la desambiguación [16]. El objetivo de la relación semántica es ponderar las asociaciones semánticas entre pares de conceptos, mientras que el objetivo de la desambiguación de entidad y palabras clave es vincular cadenas en el texto con los conceptos correspondientes, donde para hacer esto se utilizaron varios algoritmos de similitud.

Nguyen et al. [25] revisan dos medidas, SimRank y PageRank, y se investiga su idoneidad y rendimiento para calcular la similitud entre recursos en grafos RDF e investigan su uso para alimentar un sistema de recomendación basado en contenido, esto tomando en cuenta que los sistemas de recomendación basados en el contenido utilizan la noción de similitud entre los elementos; la selección de la medida de similitud basada en el grafo correcto es de suma importancia para construir un motor de recomendación efectivo.

Otra aplicación de la similitud la encontramos en el trabajo de Calado et al. [8], donde evalúan cómo se puede usar la estructura de enlace de la Web para determinar una medida de similitud apropiada para la clasificación de documentos. Experimentan con cinco medidas de similitud diferentes y determinan su adecuación para predecir el tema de una página web. Las pruebas realizadas en un directorio web muestran que la información del enlace por sí

sola permite clasificar los documentos con una precisión promedio del 86 %. Además, cuando se combina con un clasificador tradicional basado en texto, la precisión aumenta a valores de hasta 90 %. Debido a que las medidas propuestas en este artículo son sencillas de calcular, proporcionan una solución práctica y efectiva para la clasificación de documentos de la Web y las tareas de recuperación de información relacionadas.

Araujo et al. [2] intentan resolver el problema de interconexión de dos grafos. Los métodos manuales basados en reglas son la solución más efectiva para el problema, pero requieren que los editores de datos humanos calificados pasen por un proceso laborioso, propenso a errores y que requiera mucho tiempo para describir manualmente las instancias de mapeo de reglas entre dos conjuntos de datos; se propone SERIMI que hace coincidir las instancias entre un conjunto de datos de origen y uno de destino, sin conocimiento previo de los datos, el dominio o el esquema de estos conjuntos de datos.

En cuanto a consultas SPARQL, Zheng et al. [39] proponen un método sistemático para extraer diversos patrones de estructura semánticamente equivalentes. Además incorporan similitudes estructurales y semánticas en su método para encontrar consultas equivalentes.

3.4. Investigaciones en Wikidata

A continuación se describen investigaciones que han trabajado con el dataset de Wikidata. Con esto recopilamos información de cómo se han manejado estos datos: ¿Qué problemas han encontrado? y ¿Qué recomendaciones existen para trabajar con dichos datos?

Un estudio cercano al nuestro es medir la completitud de la información en Wikidata [5], donde se intenta medir qué tan completa es la información en esta base de datos; para esto se mide la cantidad de información que hay para cada entidad respecto a entidades similares, es decir qué tanta información hay de Alexis Sanchez (Futbolista Chileno) respecto a los datos de los otros jugadores en Wikidata. Si bien en este trabajo se utiliza el concepto de entidades similares, lo hace de una manera básica; se basa solamente en que dos entidades son similares si son del mismo tipo (Países con Países, Ciudades con Ciudades), y si son de tipo Humano, dos entidades son similares si son de la misma profesión (poetas con poetas, futbolistas con futbolistas, etc).

Siempre enfocado en evaluar la completitud de Wikidata, Luggen et al [21] se proponen un método capaz de estimar la integridad de una clase, tal como se define mediante un esquema u ontología, por lo tanto, se puede utilizar para responder preguntas como ¿La base de conocimientos tiene una lista completa de todas las Marcas de cerveza | Volcanes | Consolas de videojuegos? Como caso de uso, se enfocan en Wikidata, que plantea desafíos únicos en términos del tamaño de su ontología, el número de usuarios que pueblan activamente su grafo y su naturaleza extremadamente dinámica.

Un trabajo interesante a considerar por la forma de manejo de datos, realizado por Saorín et al. [27], tienen como objetivo explorar la relación entre las categorías asignadas a los artículos de Wikipedia con la descripción y metadatos generados en Wikidata. Se utiliza

la categorización de artículos de Wikipedia para enriquecer la descripción de entidades en Wikidata. Para ello se propone procesar los literales de las categorías mediante técnicas de procesamiento de lenguaje natural (PLN) estableciendo patrones que permitan identificar tanto propiedades como entidades o valores con los que se puedan construir declaraciones para una entidad.

3.5. Aporte de esta Investigación

Cada una de las secciones antes mencionadas tiene un valor de importancia para nuestro estudio, ya que pretendemos aplicar algunas de las medidas de similitud globales descritas en la Sección 3.1, ver sus fortalezas y debilidades al ser aplicadas en Wikidata, usando parámetros de comparación análogos a las investigaciones descritas en la sección 3.2, y en base a esto proponer nuevas medidas de similitud que puedan superar las debilidades de las actuales y conservar sus fortalezas, y por último comparar estas medidas propuestas, con los servicios recomendadores, una de las aplicaciones mostradas en la Sección 3.3.

Capítulo 4

Algoritmos de Similitud Propuestos

A continuación se describen las medidas de similitud que se proponen en este trabajo, para ser aplicadas en Wikidata; cada una de las siguientes medidas, representa la evolución de una iteración de aplicación y evaluación de las medidas, intentando superar las debilidades encontradas.

4.1. Preliminares

Dado un grafo dirigido $G = (V, E)$ donde V es un conjunto de vértices (o nodos) y $E \subseteq V \times V$ es un conjunto de arcos dirigidos (o aristas dirigidas), una medida de similitud global es una relación:

$$\sigma : V \times V \rightarrow [0, \infty) \quad (4.1)$$

Lo que se busca es asignar un valor a cada par de vértices del grafo; este valor representa la similitud entre dos vértices. Para fines de este estudio, el valor de similitud no será negativo, ya que la interpretación de una similitud negativa implica otro enfoque de estudio. Además, en este trabajo, un valor mayor indica un par de nodos más similares.

Otras definiciones necesarias para entender las medidas propuestas en este trabajo son:

- $N(v)$: Es el conjunto de vecinos de v en G , formalmente,

$$N(v) = \{x \in V \mid (v, x) \in E\}$$

.

- $deg(n)$: Es el número de nodos que tiene como vecino a n en G , formalmente:

$$deg(n) = |\{v \in V \mid n \in N(v)\}|$$

4.2. Conteo de Vecinos Comunes

La primera medida propuesta, intenta medir la similitud de dos entidades contando las características comunes entre estas dos entidades, esto tomando el hecho que una arista (s, p, o) en RDF representa una propiedad o característica del sujeto s .

Al observar la información mostrada en la Figura 1, si tomamos solamente los nodos para encontrar los vecinos comunes, estaremos obviando el hecho de que Clint Eastwood es director y actor en dos de las películas, pero en la tercera sólo es actor. Por lo tanto, tomaremos por cada triple (s, p, o) el par (p, o) como vecino, de esta manera usamos la información de la arista, quedando la información representada como se muestra en la Figura 4.1:

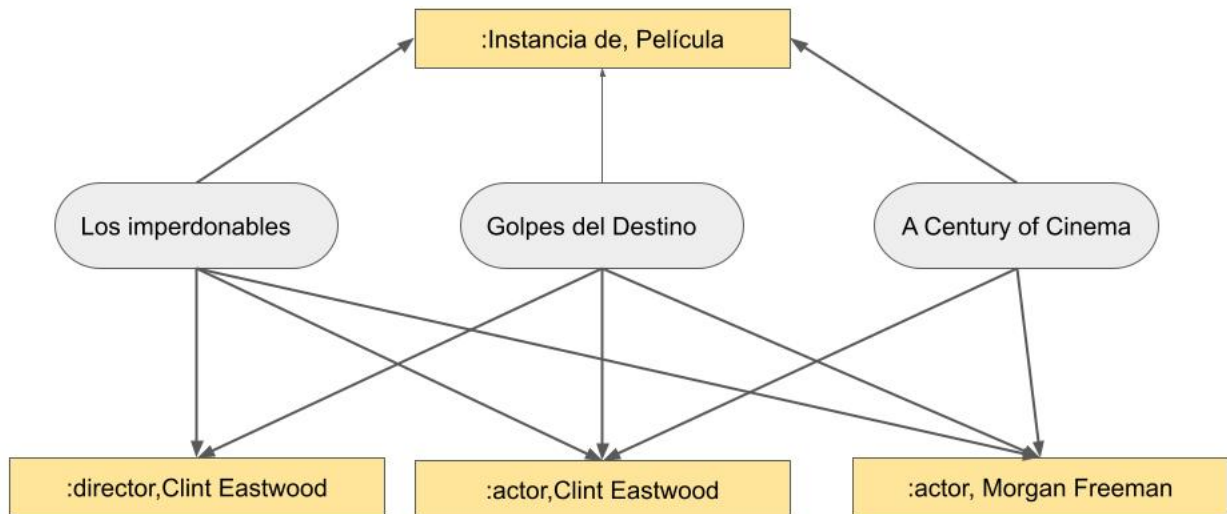


Figura 4.1: Nueva representación de la información

De esta forma no sólo tomamos el hecho de la participación de un sujeto, además diferenciamos el rol de dicho sujeto en la filmación de la película.

Luego de hacer este ajuste, la primera métrica de similitud que se aplicó fue contar la cantidad de elementos comunes que tiene cada par de entidades.

Formalmente :

$$\sigma_{nc}(v_1, v_2) = |N(v_1) \cap N(v_2)| \quad (4.2)$$

En la Figura 4.2 se muestra la vecindad de la película *Los Imperdonables*:

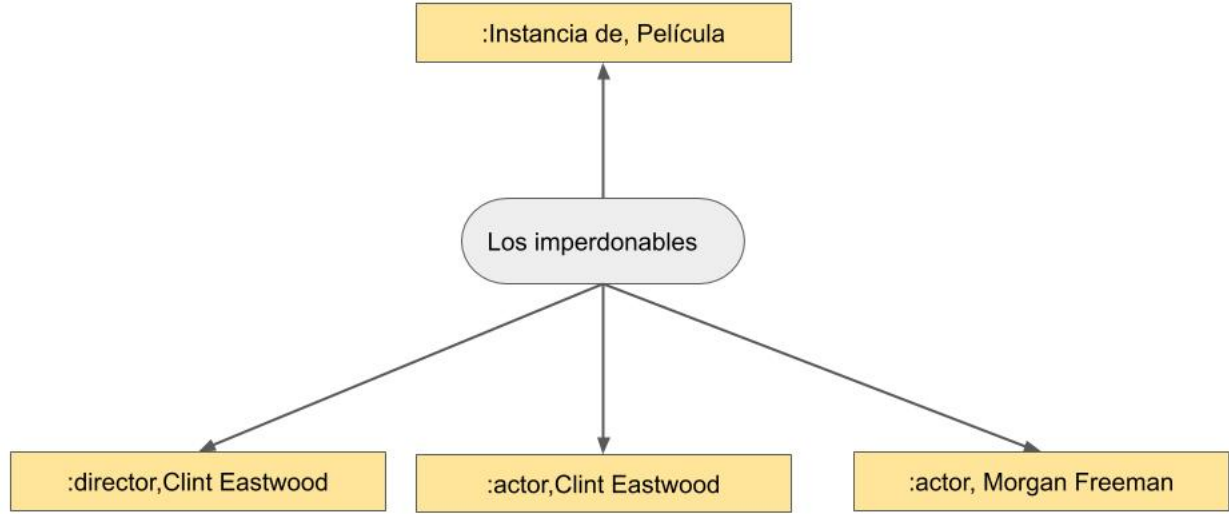


Figura 4.2: Vecindad de Los Imperdonables

Los 4 nodos que tienen como vecinos *Los Imperdonables*, los tiene de igual manera la entidad *Golpes del Destino*, por lo que la similitud entre estas dos entidades es igual a 4. De esta manera, se calcula la similitud entre cada par de entidades; luego se ordena de mayor a menor, ya que un valor alto significa más similitud.

Analizando un poco, la medida de conteo de vecinos trata a todos los nodos iguales, sin embargo, existen nodos que pueden ser muy comunes como ser (*país de origen, Estados Unidos*) ya que la mayoría de las películas son producidas en Estados Unidos, mientras que el nodo (*premio recibido, Oscar a mejor película*) no es tan común entre películas, por lo que se debe buscar una medida que haga distinción entre nodos poco frecuentes y nodos muy comunes.

4.3. Suma Probabilística de Vecinos Comunes

Como se resaltó en la sección anterior, existen nodos que son muy comunes en las vecindades de las entidades. Denotamos la probabilidad de que n aparezca en la vecindad de un nodo aleatorio como $P(v) = \frac{\deg(v)}{|V|}$.

En lugar de simplemente contar a los vecinos que dos nodos tienen en común, proponemos esta nueva medida, que define la similitud de dos vértices v_1 y v_2 como la probabilidad de que un nodo aleatorio v no tenga todos los vecinos individuales que v_1 y v_2 tienen en común.

Mostramos la deducción de la medida; lo ejemplificaremos con la información de la Figura 4.1. Si definimos X como el evento de que una entidad tenga como director a *Clint Eastwood*, a Y el evento que una entidad tenga al mismo *Clint Eastwood* como actor, y Z como el evento de que una entidad tenga como actor a *Morgan Freeman*. Entonces sus probabilidades son:

$$P(X) = \frac{2}{7} \quad (4.3)$$

$$P(Y) = \frac{3}{7} \quad (4.4)$$

$$P(Z) = \frac{3}{7} \quad (4.5)$$

Luego si asumimos que X, Y, Z son eventos independientes tendremos que:

$$P(X \cap Y \cap Z) = P(X) \times P(Y) \times P(Z) = \frac{18}{343} \quad (4.6)$$

Lo que se interpretaría como: La probabilidad de que una entidad tenga a *Clint Eastwood* como actor y director, y además tenga *Morgan Freeman* como actor es $\frac{18}{343}$, que es equivalente a la probabilidad de que una entidad tenga en su vecindad, las cosas que tiene en común la películas *Los imperdonables* y *Golpes del Destino*.

Luego si tomamos que $N(v_1) \cap N(v_2) = \{n_1, \dots, n_k\}$ entonces la similitud de v_1 y v_2 estará dado por:

$$\sigma_{NS}(v_1, v_2) = \neg P(n_1 \cap \dots \cap n_k) = 1 - \prod_{i=1}^k P(n_i) \quad (4.7)$$

Esto supone que tener vecinos individuales son eventos independientes. La idea detrás de esta medida es que cuantos más vecinos compartan un par de nodos, y cuanto más raros sean esos vecinos, menor será la probabilidad de que un nodo aleatorio tenga todos esos vecinos y, por lo tanto, mayor será la medida de similitud.

Sin embargo, como se expresó anteriormente, este análisis asume que los eventos son independientes, pero en Wikidata podemos encontrar casos en que la ocurrencia de un (p, o) es dependiente de otro.

En la Figura 4.3 se muestra un caso de dependencia de vecinos, ya que el par *(parte de, Latinoamérica)* es dependiente del par *(lenguaje Oficial, Español)*.

Obviamente este tipo de dependencia cambia según el dominio y cómo se modelaron los datos, por lo que la búsqueda de estas dependencias es un trabajo que se debe hacer antes

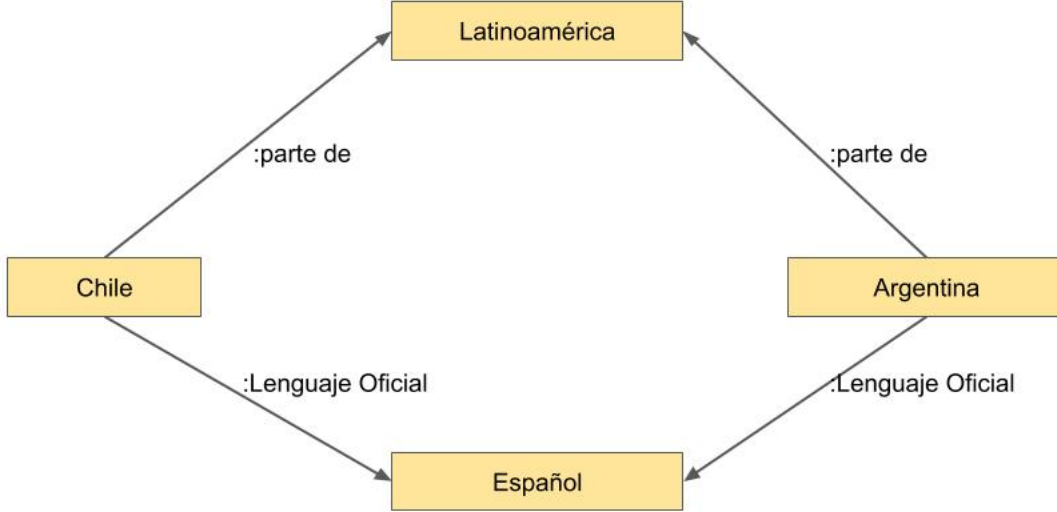


Figura 4.3: Ejemplo de Dependencia de (p, o)

de calcular la probabilidad de la intersección de eventos, de lo contrario se estaría tomando eventos dependientes como si no lo fueran, generando un error en la métrica de similitud.

4.4. Conteo de Intersección

En presencia de datos redundantes o interdependientes, que, como hemos argumentado anteriormente, son comunes en el caso de las bases de datos de grafos, la medida anterior terminaría sobreestimando la similitud de los pares de nodos según sus propios criterios de diseño.

Buscando robustez ante la dependencia de eventos descrita en la sección anterior, se propone una medida de similitud definida como la rareza del vecindario común entre dos entidades. Es decir, para dos nodos v_1 y v_2 la similitud de estos será igual a la probabilidad de que un nodo aleatorio que no sea v_1 ni v_2 no tenga todos los vecinos que v_1 y v_2 tengan en común. Primero vamos a generalizar la definición de grado a un conjunto de nodos, si tenemos a $K \subseteq V$ un conjunto de nodos, entonces definimos $deg(K) = |\{v \in V \mid K \subseteq N(v)\}|$; deg representan el número de nodos que tiene en su vecindad incluido a K . Ahora definimos la medida de similitud de conteo de intersección como:

$$\sigma_{NR}(v_1, v_2) = 1 - \frac{deg(N(v_1) \cap N(v_2)) - 2}{|V| - 2} \quad (4.8)$$

Restamos 2 para excluir v_1 y v_2 del recuento (implícitamente se asume que hay más de dos nodos en el grafo). A diferencia de la medida anterior, ahora si tenemos vecinos interdependientes en $N(v_1) \cap N(v_2)$, entonces la rareza del vecindario tendrá esto en cuenta ya que muchos otros nodos también tienen esos vecinos interdependientes juntos, reduciendo la similitud anterior.

En la ecuación anterior se está midiendo qué tan común es el conjunto de vecinos que comparten dos entidades.

Si aplicamos esta medida a la información de la Figura 4.1, tendríamos que la similitud de las películas *Los imperdonables* y *Golpes del Destino* sería igual a 1, ya que no hay otro nodo distinto a estas dos, que tenga como vecinos a $\{director:Clint Eastwood, actor:Clint Eastwood, actor:Morgan Freeman, instanciaDe: Película\}$ en su totalidad.

4.5. SimRank

Como se mencionó en el capítulo de trabajos relacionados, una de las medidas de similitud más populares para grafos es SimRank [17], la cual usa el principio de que **dos objetos son similares si están relacionados con objetos similares**. Para poder definir esta medida necesitamos agregar notación adicional. Sea $N^-(v) = \{n \in V \mid (n, v) \in E\}$ el conjunto de nodos con aristas entrantes a v .

El valor de similitud R se calcula con la siguiente fórmula:

$$R_{k+1}(a, b) = \frac{C}{|N^-(a)||N^-(b)|} \sum_{i=1}^{|N^-(a)|} \sum_{j=1}^{|N^-(b)|} R_k(N_i^-(a), N_j^-(b)) \quad (4.9)$$

Donde C es un coeficiente de decaimiento de transferencia de similitud, (normalmente $C = 0,80$), y $N_i^-(a)$ representa el elemento i del conjunto $N^-(a)$.

Intuitivamente, la similitud de a y b (donde $a \neq b$) depende de la suma de la similitud por pares de sus vecinos entrantes, normalizada por la puntuación que tendrían si todos los vecinos entrantes tuvieran una similitud por pares de 1 ($|N^-(a)| \cdot |N^-(b)|$), multiplicada por un parámetro de decaimiento.

Se puede observar que la fórmula 4.9 es recursiva; esta fórmula empieza en el siguiente caso base:

$$R_0(a, b) = \begin{cases} 0 & \text{si } a \neq b \\ 1 & \text{si } a = b \end{cases} \quad (4.10)$$

En el paper original [17] se establece que se puede encontrar k donde los valores de similitud se estabilizan, sin embargo se recomienda un valor máximo de $k = 5$.

4.6. Comparaciones de las Medidas Propuestas

Nuestra predicción es que la medida más robusta es la medida de Conteo de Intersección, ya que esta usa aspectos favorables de las dos medidas anteriores, además de intentar superar las debilidades encontradas. Sin embargo el mayor de los objetivos es vencer en rendimiento y calidad de resultados a SimRank, el algoritmo base que se utilizó para compara los resultados.

Cabe destacar que hay diferencias entre nuestras medidas y la medida SimRank. Primero, la medida SimRank es recursiva mientras que nuestras medidas no son recursivas. Segundo, nuestra medida considera la información de las aristas (es decir sus etiquetas) mientras que SimRank, por defecto, no considera esta información dado que ha sido definido en el contexto de grafos dirigidos sin etiquetas. Tercero, nuestra medida sólo calcula la similitud entre pares de nodos con algún vecino común (se puede observar que el caso de nodos sin algún vecino en común, las primeras tres medidas devuelven un puntaje de 0), mientras que SimRank es una medida global que tiene que calcular la similitud entre todos los pares de nodo del grafo. Dados los puntos segundo y tercero, esperamos que nuestras medidas sean más escalables que SimRank. Haremos una comparación experimental con SimRank en la sección 6.5; ahora analizamos el costo de las medidas.

Para cumplir dicho objetivo debemos partir conociendo ¿Cuál es el costo computacional de una implementación de las medidas propuestas? Para esto, en el siguiente capítulo se muestran las implementaciones de las medidas, describiendo cómo se abordó el tema de escalabilidad y se presenta una solución de acotamiento del costo computacional. La solución general que proponemos es aproximar las medidas, sólo considerando vecinos que tiene, como máximo, k nodos asociados.

Capítulo 5

Implementación de las Medidas

En este capítulo hablaremos de la implementación de las medidas propuestas en el capítulo anterior; se usará Spark para facilitar la computación distribuida y eficiente, ya que el mayor desafío es la escalabilidad dados los costos previamente analizados.

Es importante resaltar que la información en Wikidata, se guardan en triples (s, p, o) por lo que las siguientes implementaciones, se basan en un archivo que contiene triples.

5.1. Extractor de Tuplas

En los experimentos, calcularemos las medidas de similitud sobre conjuntos de entidades del mismo tipo por lo que al inicio se creó una clase que extrae de Wikidata todas las triples (s, p, o) donde s es una entidad de una clase específica; por ejemplo películas, países, modelos de autos, entre otros.

Para lograr identificar los sujetos de una clase, se usa el predicado P31, que significa **instancia de** en Wikidata.

```

1 extract_triples(wikipedia_file, domain, directory){
2     % Extraer sujetos.
3     subjects=[ ]
4     input_file= open(wikipedia_file)
5     for tr in input_file:
6         if tr[1]==P31 and tr[2]==domain: % P31 indica instancia de
7             subjects.add(tr[0])
8         end if
9     end for
10    % Extraer triples
11    output_file = open(directory + "result.txt")
12    for tr in input_file:
13        if subjects.contains(tr[0]):
14            output_file.write(tr)
15        end if
16    end for
17    close(input_file)
18    close(output_file)
19 }

```

El archivo final es utilizado para aplicar los distintos algoritmos de similitud.

5.2. Conteo de Vecinos Comunes

Se utilizó Spark para realizar el conteo de los vecinos comunes, por lo que el siguiente pseudocódigo se usarán las funciones descritas en la sección 2.4.

```

1 contar_vecinos(result_file, k){
2     input_file = open(result_file);
3     <s,<p,o>> triple = map(line in :result_file);
4     <<p,o>,s> triple_inv = triple.swap();
5     <<p,o>,s[ ]> group = triple_inv.groupByKey();
6     <<p,o>,s[ ]> fil = group.filter(len(s[ ]) <= k)
7     <<s1,s2>,1> cnt = fil.flatMapToPair(
8         for s1 in s[ ]:
9             for s2 in s[ ]:
10                if s1<s2:
11                    return <<s1,s2>,1>
12                else:
13                    return <<s2,s1>,1>
14    );
15    <<s1,s2>,n> redu = cnt.reduceByKey((a,b)->a+b);
16    redu.saveAsTextFile()
17 }

```

Recordando que según lo explicado en la sección 3.1, se agrupa el predicado y el objeto como nodo, por lo que en la línea 2 se usa el par $(sujeto, (predicado, objeto))$.

En la línea 5 se agrupan los sujetos que tiene como vecino al nodo (p, o) . Luego en la línea 6 filtramos los (p, o) que son más comunes; este filtro es necesario ya que en la línea 7, se crean

pares sujeto a sujeto, y esto tiene costo cuadrático por arista, por lo que podría tomar tiempo y memoria considerablemente. Sin embargo, si se escoge un k grande, se estarán filtrando nodos muy comunes, que al final sumarán a la vecindad de muchos sujetos, por lo que no son tan relevantes en estas medidas de similitud. Este valor de k sirve como parámetro de aproximación de la medida; un valor menor implicará una computación más eficiente pero menos precisa. Formalmente estamos eliminando los nodos (p, o) que tienen grado mayor a k . En la sección 6,3, veremos un análisis sobre el efecto de este parámetro en la medida de similitud de vecinos comunes.

5.3. Suma Probabilística de Vecinos Comunes

Según lo explicado en el capítulo anterior, la métrica de Suma Probabilística de Vecinos Comunes establece, que la similitud entre v_1 y v_2 será la probabilidad de que una entidad v_3 no tenga en su vecindario todos los elementos que tienen v_1 y v_2 en común, considerando cada vecino como un evento independiente.

```

1 sum_prob(result_file , k){
2   input_file = open(result_file);
3   <s,<p,o>> triple = map(line in :result_file);
4   n = triple.Keys();
5   <<p,o>,s> triple_inv = triple.swap();
6   <<p,o>,s[ ]> group = triple_inv.groupByKey();
7   <<p,o>,s[ ]> filt = group.filter(len(s[ ]) <= k);
8   <<s1,s2>,c/n> count = filt.flatMapToPair(
9     c=len(s[ ])
10    for s1 in s[ ]:
11      for s2 in s[ ]:
12        return <<s1,s2>,c/n>
13  );
14  <<s1,s2>,prob> redu = count.reduceByKey((a,b)->a*b);
15  <<s1,s2>,prob> redu_neg = redu.map(prob->1-prob);
16
17  redu_neg.saveAsTextFile()
18 }
```

En la línea 4 se calcula el grado de la vecindad. Las líneas 5, 6 y 7 son iguales al algoritmo anterior. Sin embargo, en las líneas del 8 al 14, se calcula la probabilidad de encontrar un sujeto que tenga el mismo (p, o) que $s1, s2$.

En la línea 16 se calcula la probabilidad de encontrar un sujeto que tenga todos los (p, o) que tiene en común $s1, s2$; por último se resta la probabilidad de 1 y se guardan los resultados.

5.4. Conteo de Intersección

Siguiendo la idea del capítulo anterior, esta implementación busca contar la cantidad de entidades que tengan como vecinos al menos todos los vecinos que tienen v_1 y v_2 en común.

Esta idea se desarrolló en dos pasos:

1. Contar la cantidad de elementos de la intersección de cada par de sujetos : $|N(v_1) \cap N(v_2)|$.
2. Contar la cantidad de elementos de la intersección de cada triple de sujetos : $|N(v_1) \cap N(v_2) \cap N(v_3)|$.

Partiendo del hecho de que si $|N(v_1) \cap N(v_2)| = |N(v_1) \cap N(v_2) \cap N(v_3)|$ entonces $N(v_1) \cap N(v_2) \subseteq N(v_3)$, que podemos usar este método para encontrar las entidades v_3 cuyos vecinos cubren todos los vecinos comunes de v_1 y v_2

Para encontrar el valor de $|N(v_1) \cap N(v_2)|$ se utilizó el código de la función *contar vecinos* descrita anteriormente, con la única diferencia, que para fines de esta medida se guarda en otro archivo la cantidad de nodos $|V|$, que luego será necesaria como parámetro en la función *filter inter*. Luego, se sigue con el siguiente método, que cuenta los (p, o) que tienen en común 3 entidades cualesquiera.

```

1 contar_interseccion_triple(result_file , k){
2     input_file = open(result_file);
3     <s,<p,o>> triple = map(line in :result_file);
4     <<p,o>,s> triple_inv = triple.swap();
5     <<p,o>,s[ ]> group = triple_inv.groupByKey();
6     <<p,o>,s[ ]> filt = group.filt(len(s[ ])<=k)
7     <<s1,s2,s3>,1> count = filt.flatMapToPair(
8         for s1 in s[ ]:
9             for s2 in s[ ]:
10                 for s3 in s[ ]:
11                     if(s1<s2 & s1!=s3)
12                         return <<s1,s2,s3>,1>
13     );
14     <<s1,s2,s3>,cont> redu = count.reduceByKey((a,b)->a+b);
15     redu.saveAsTextFile()
16 }
```

Otro punto importante es la decisión del último ciclo for; en un inicio se tomó que $s1 < s2 < s3$ para que cada posible triple de sujetos se tomara solo una vez, sin embargo con esto se descartaba los pares de sujetos $s1, s2$ que sus vecinos en común no los tiene nadie más, siendo el caso más importante en este método, por que significa que las cosas en común que tiene $s1, s2$ no las tiene nadie más.

El siguiente método `filter_inter` carga los dos resultados encontrados anteriormente, la doble intersección se lee en la línea 4, mientras que la triple intersección es cargada en la línea 5, dejando en ambos casos como llave el primer par de entidades.

```

1 filter_inter(doble_file, triple_file, V){
2   input_file_2 = open(doble_file);
3   input_file_3 = open(triple_file);
4   <<s1,s2>,m> doble = map(line in :input_file_2);
5   <<s1,s2>,<s3,n>> triple = map(line in :input_file_3);
6   <<s1,s2>,<s3,m,n>> join = doble.join(triple);
7   <<s1,s2>,<s3,m,n>> filt = join.filter(m==n);
8   <<s1,s2>,c> map = filter.flatMapToPair(
9     if s2==s3:
10       return <<s1,s2>,0>;
11     else:
12       return <<s1,s2>,1>;
13
14   );
15
16   <<s1,s2>,cont> redu = count.reduceByKey((a,b)->a+b);
17   <<s1,s2>,cont> redu_final = redu.map(cont->1- (cont-2)/(V-2));
18
19   redu_sort.saveAsTextFile()
20 }

```

En la línea 6 y 7 se hace un join entre los archivos, obteniendo los tres sujetos $s1, s2, s3$ tal que $|N(s1) \cap N(s2)| = |N(s1) \cap N(s2) \cap N(s3)|$.

En las líneas del 8 al 14, se mapean los resultados del filtro. Asignamos 0 si $s2 = s3$; recordamos que esto puede pasar ya que la condición del último ciclo for del método `contar_interseccion_triple`, permite esto para contar los casos en donde para un par de sujetos $s1, s2$ no exista un $s3$ que tenga los vecinos comunes de $s1, s2$.

El otro caso es que $s3$ sea diferente a $s2$; a este le asignamos 1, lo cual significa que encontramos un $s3 \neq s2 \neq s1$ que tiene en su vecindad todos los vecinos comunes de $s1, s2$.

Luego en la línea 16 se reduce por llave los resultados mapeados en el trabajo anterior, y se ordenan de manera ascendente.

5.5. Costo de las Implementaciones

Para simplificar la explicación tomaremos que $n = |V|$

- **Conteo de Vecinos Comunes:** Esta implementación tiene costo $O(n^3 \log(n))$ sobre el número de entidades; si ordenamos el conjunto de relaciones E , y tenemos el peor caso que sería tener un grafo donde todos se relacionen con todos, tendríamos un conjunto de pares de tamaño $|n^2|$, luego tendremos que ir contando los vecinos de cada nodo, es decir pasar $|n|$ veces, por lo que tendremos un costo final de $O(n^3)$; que luego debemos ordenar, para tener un costo final de $O(n^3 \log(n))$.

- **Suma Probabilística de Vecinos Comunes:** Debido a que esta medida tiene una implementación similar a *Conteo de Vecinos Comunes* entonces el costo $O(n^3 \log(n))$ se mantiene, siguiendo el análisis antes descrito.
- **Conteo de Intersección:** Siguiendo la idea anterior pero tomando la implementación mostrada de la función *contar intersección triple*, se puede observar tres ciclos for anidados, por lo que esta parte de la implementación tiene un costo n^3 , ahora tomando el peor caso antes descrito ($|E| = n^2$), tendremos un costo final de $O(n^4 \log(n))$

Tomando en cuenta el valor de aproximación k que recibe cada medida, la complejidad de los primeros dos algoritmos es de $O(m \log m + nk^2 \log n)$, donde m es el número de aristas en el grafo. Esta complejidad considera que tenemos que ordenar todas las aristas para agruparlas por sujeto, y luego para cada sujeto, generar al máximo k^2 pares de vecinos que tendremos que ordenar. En el caso de la tercera medida, la complejidad es de $O(m \log m + nk^3 \log n)$, considerando que tenemos que generar, para cada nodo, al máximo k^3 tripletas de vecinos. Si consideramos que el valor de k es un constante, entonces la complejidad de los tres algoritmos será igual: $O(m \log m)$ (asumiendo que cada nodo tiene al menos una arista incidente).

Este resultado demuestra que el factor más costoso asociado con los tres algoritmos es el grado máximo de un nodo en el grafo, así que limitando eso puede ser una buena forma de aproximar los resultados.

Capítulo 6

Evaluación

En este capítulo se describen las pruebas realizadas para evaluar el rendimiento y calidad de los resultados de las medidas propuestas; además se describen los datos utilizados para realizar dichas pruebas.

6.1. Descripción de Datos

En la Tabla 6.1 se describen los dominios seleccionados para aplicar las medidas de similitud propuestas; esta primera selección son triples (s, p, o) donde s es una instancia de la clase seleccionada. En las secciones siguientes se describen nuevamente los dominios donde el objeto también puede ser una entidad del dominio.

Se escogieron los dominios de Películas y Album de música, con el objetivo de comparar los resultados que arrojan las medidas con algunos servicios de recomendación, donde dicho servicio recomienda entidades similares; en la Tabla 6.1 se muestran las entidades de cada dominio, los predicados y objetos relacionados a estas entidades.

Los demás dominios se escogieron para ver el comportamiento de las medidas en otros dominios con características diferentes.

Domino	Entidades	Predicados	Objetos	Triples
Películas	765	896	256456	365123
Modelos de Autos	38	89	3115	356416
Países	179	905	196318	216774
Universidades	842	787	83583	94599
Álbum de Música	653	936	165456	189653

Tabla 6.1: Descripción de Dominios Utilizados.

6.2. Tiempos en Calcular las Medidas

Primero presentamos el rendimiento de las medidas respecto al valor de k (el umbral superior del grado de los nodos considerados) y el tiempo que se tomó en calcular los resultados.

En cada dominio se incrementó el valor de aproximación k y se medía el tiempo en segundos que tomaba calcular los resultados por cada incremento de k .

Es importante resaltar que la medida de Vecinos Comunes como se describió anteriormente produce una cantidad considerable de empates, y la medida de Suma Probabilística intenta superar este problema calculando la probabilidad de cada vecino.

Valor de k	Vec. Comunes	Sum. Proba	Conte. Inter
4	162	183	433
8	164	200	503
16	175	237	732
32	200	262	1989
64	283	301	7303
128	417	367	11514
256	547	447	20736
512	1162	737	

Tabla 6.2: Tiempo en segundos para Películas

En la Tabla 6.2 se muestra el tiempo en segundos que tomó calcular cada una de las medidas con diferentes valores de k en el dominio de Películas; para la medida Conteo de Intersección y el valor $k = 512$ no se pudo calcular el tiempo, ya que en este dominio la cantidad de entidades exigían una gran cantidad de memoria.

En la Figura 6.1 se representa gráficamente los valores de la Tabla 6.2; para poder observar bien los resultados del tiempo, se aplicó una escala logarítmica a los valores del tiempo en el eje y .

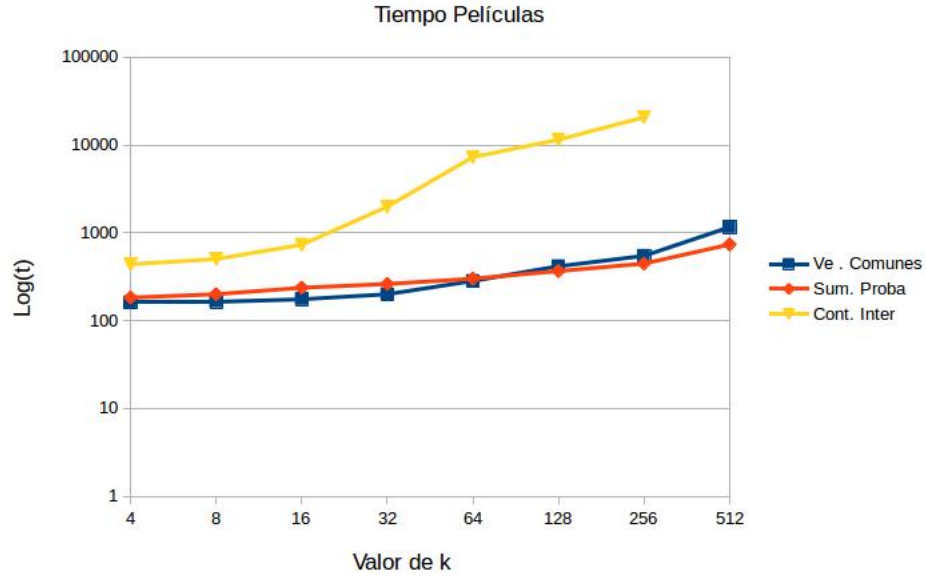


Figura 6.1: Tiempo en Películas

En la Tabla 6.3 se muestran los tiempos en Universidades; en este dominio, se logró correr todas las pruebas.

Valor de k	Vec. Comunes	Sum. Proba	Conte. Inter
4	145	159	413
8	149	159	411
16	149	160	412
32	152	160	420
64	147	159	431
128	145	161	491
256	147	160	697
512	149	162	710

Tabla 6.3: Tiempo en segundos para Universidades

La Figura 6.2 es la representación gráfica de los valores de la tabla anterior, en este caso no se aplicó la escala logarítmica. Se puede notar que los valores de los tiempos son más bajos que lo se tomó en películas; esto se puede explicar por el número de triples que tiene los dominios.

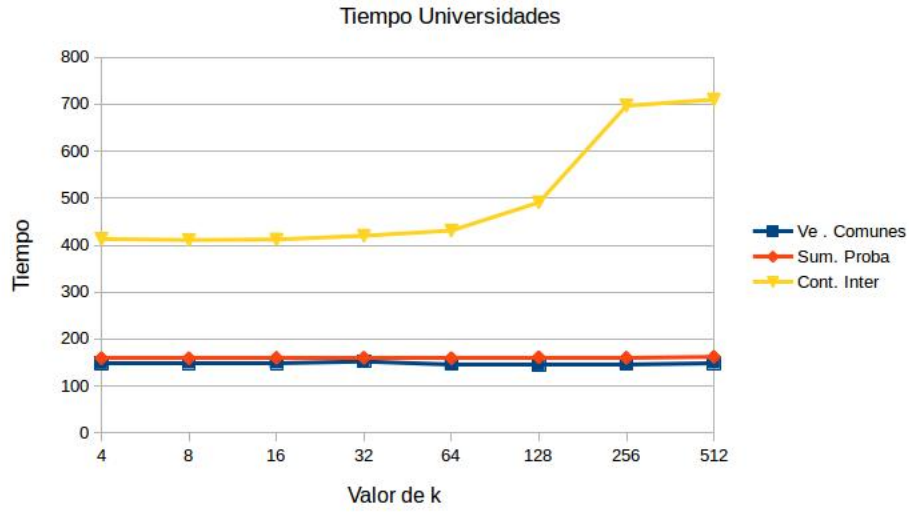


Figura 6.2: Tiempo en segundos para Universidades

En la Tabla 6.4 se muestran los segundos que se tardó en calcular los resultados en el dominio de países; igual que en Universidades, por el número de triples, se logró correr todas las pruebas. En la Figura 6.3 se encuentran graficados los valores de la tabla anterior.

Valor de k	Vec. Comunes	Sum. Proba	Conte. Inter
4	147	160	413
8	145	160	416
16	146	162	414
32	146	160	429
64	152	161	461
128	147	162	520
256	146	162	819
512	147	162	812

Tabla 6.4: Tiempo en segundo para Países

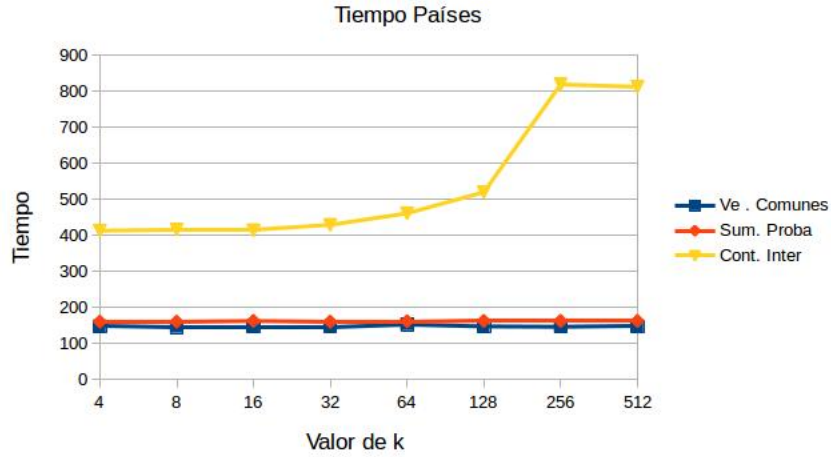


Figura 6.3: Tiempo en segundo para Países

En la Tabla 6.5 se muestran los tiempos en segundos demorados en calcular los resultados en Álbumes de Música; al igual que en Películas, por la cantidad de entidades, no se pudo realizar la prueba $k = 512$ para la medida de Conteo de Intersección, recordando que esta medida tiene un costo alto sobre el valor de k .

Valor de k	Vec. Comunes	Sum. Proba	Conte. Inter
4	175	183	439
8	185	200	503
16	195	237	985
32	200	262	1989
64	283	350	7303
128	417	459	12365
256	547	568	18056
512	1535	856	

Tabla 6.5: Tiempo en segundos para Álbumes de Música

En la Figura 6.4 se muestra la gráfica de los valores de tiempo mostrado en la Tabla 6.5 esta vez con una escala logarítmica.

Como se observa en los gráficos mostrados en esta sección, es claro que la medida Conteo de Intersección siempre toma más tiempo que las dos otras medidas, mientras que las medidas Vecinos Comunes y Suma Probabilística casi toman el mismo tiempo en calcularse. Sin embargo esto es algo esperado, debido a que como se explicó en la sección anterior Conteo de Intersección tiene un costo n^4 en el número de entidades, mientras que las otras dos medidas tienen un costo cúbico.

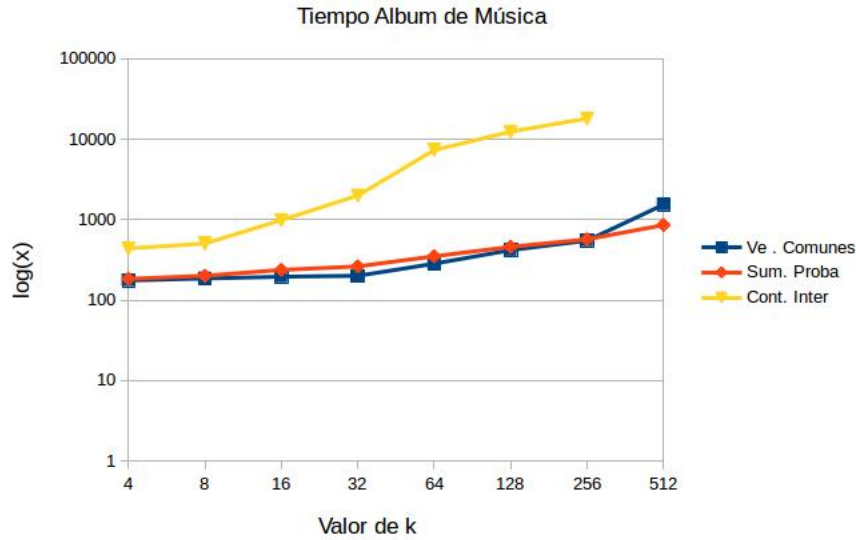


Figura 6.4: Tiempo en segundos para Álbumes de Música

6.3. Efecto del valor de k en las medidas.

Estas pruebas tienen como objetivo, evaluar qué efecto tiene el parámetro k en las medidas de similitud propuestas; se pretende responder a las preguntas ¿Qué tanto afecta tener un k bajo? ¿Es necesario tener un k alto para tener buenos resultados?

Las pruebas consisten en calcular las medidas de similitud con un $k = 512$ (valor más alto de k que se pudo calcular), luego calcular nuevamente las medidas de similitud con un valor más bajo de k , y ver qué correlación hay entre estos dos resultados.

Seguidamente se reduce más el valor de k , y se vuelve a evaluar los resultados respecto a los resultados obtenidos con $k = 512$, es decir ir reduciendo el valor de k y comparar cada uno de los resultados con los resultados del mayor k que se pudo calcular.

Si bien las medidas de similitud devuelven un valor numérico que representan la similitud de las entidades, al ser medidas globales, los resultados normalmente se ordenan por el valor de su similitud, es decir que la pregunta clave en este contexto es ¿Cuáles son las entidades más similares? y no ¿Qué tan similar son los puntajes calculados por cada medida?

Al ver los resultados como rankings, se pueden medir la correlación entre los resultados de las medidas, midiendo qué tan diferentes son los rankings que se forman con los resultados; para esto se utilizó la medida de correlación tau-b de Kendall, descrita a continuación.

6.3.1. Coeficiente de Correlación tau-b de Kendall

Cuando se estudia la relación entre variables de tipo ordinal se debe utilizar el coeficiente de correlación de rangos de Kendall [18], denominado tau de Kendall, del cual existe una

variante *tau-b*. Además, según Morales y Rodríguez [22], su aplicación tiene sentido si las variables objeto de estudio no poseen una distribución normal.

Para determinar el coeficiente de correlación de Kendall en una muestra de tamaño n pares de datos, tomados de dos variables aleatorias, digamos X e Y , el cual se denota por T_{xy} , se utiliza la expresión [30]:

$$T_{xy} = \frac{2S}{n(n+1)} \quad (6.1)$$

Dónde $S = P - M$ siendo P el número de valores positivos o “Acuerdos”, y M número de valores Negativos o “Desacuerdos”. Un Acuerdo significa que un par de resultados aparecen en el mismo orden, un ejemplo de un Acuerdo sería: si en un ranking A tiene la posición 3 y B la posición 4, y en el segundo ranking A tiene posición 2 y B tiene posición 5, entonces en ambos rankings existe un Acuerdo de que A es más alto que B .

La fórmula 6.1 se utiliza cuando no hay observaciones de empates; sin embargo al trabajar con las medidas de similitud, existe la posibilidad de que dos pares de entidades tengan el mismo valor de similitud.

Igualmente existe una forma de calcular la correlación Kendall para situaciones donde se observan empates [22].

$$T_{xy} = \frac{2S}{\sqrt{n(n-1) - T_x} \cdot \sqrt{n(n-1) - T_y}} \quad (6.2)$$

Donde

$$T_x = \sum t_x(t_x - 1) \quad (6.3)$$

$$T_y = \sum t_y(t_y - 1) \quad (6.4)$$

Siendo t el número de observaciones empatadas por cada grupo de empates. El coeficiente T_{xy} de Kendall varía de -1 a $+1$, donde -1 significa que los rankings van en el orden inverso, y 1 implica que los rankings van en el mismo orden.

6.3.2. Valor de p y la significación estadística

El valor p es una probabilidad que mide la evidencia en contra de la hipótesis nula [23]. Las probabilidades más bajas proporcionan una evidencia más fuerte en contra de la hipótesis nula.

Se utiliza p para determinar si se puede o no se puede rechazar la hipótesis nula, que indica que las variables son independientes.

Para determinar si las variables son independientes, se compara el valor p con el nivel de significancia. Por lo general, según Anderson et al. [1], un nivel de significancia (denotado como α o alfa) de 0,05 es generalmente aceptado . Un nivel de significancia de 0,05 indica un riesgo de 5 % de rechazar una hipótesis nula verdadera.

6.3.3. Resultados de las Pruebas

k	Países	Universidades	Películas	Alb. Música	Autos
4	*0,25	*0,20	*0,25	*0,22	*0,31
8	*0,35	*0,26	*0,35	*0,30	*0,37
16	*0,58	*0,53	*0,48	*0,52	*0,49
32	*0,78	*0,72	*0,56	*0,58	*0,52
64	*0,89	*0,85	*0,75	*0,70	*0,79
128	*0,95	*0,90	*0,85	*0,83	*0,86
256	*1,00	*0,99	*0,93	*0,95	*0,98
512	*1,00	*1,00	*1,00	*1,00	*1,00

Tabla 6.6: Efecto de k en Conteo de Vecinos Comunes

En la Tabla 6.6 se muestran los valores del τ -b para la métrica de Conteo de Vecinos Comunes, aclarando la siguiente notación:

- Si el valor de $p < 0,05$ en la prueba, entonces se marca con una asterisco el valor de la prueba, ejemplo *0,56 .

En la Figura 6.5 se muestran de forma gráfica los valores de la tabla anterior.

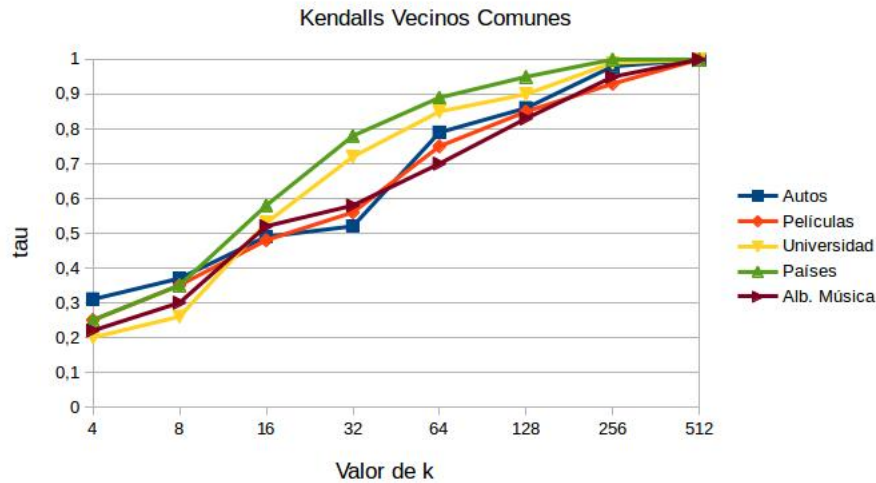


Figura 6.5: Efecto de k en vecinos comunes

En cuanto a la medida de Suma Probabilística los resultados se presentan en la siguiente tabla, y en la Figura 6.6 .

k	Países	Universidades	Películas	Alb. Música	Autos
4	*0,23	*0,29	*0,62	*0,59	*0,41
8	*0,33	*0,34	*0,63	*0,60	*0,60
16	*0,44	*0,40	*0,66	*0,75	*0,65
32	*0,79	*0,56	*0,73	*0,79	*0,69
64	*0,91	*0,76	*0,82	*0,84	*0,77
128	*0,99	*0,86	*0,88	*0,90	*0,89
256	*1,00	*0,99	*0,93	*0,95	*0,90
512	*1,00	*1,00	*1,00	*1,00	*1,00

Tabla 6.7: Efecto de k en Suma Probabilística

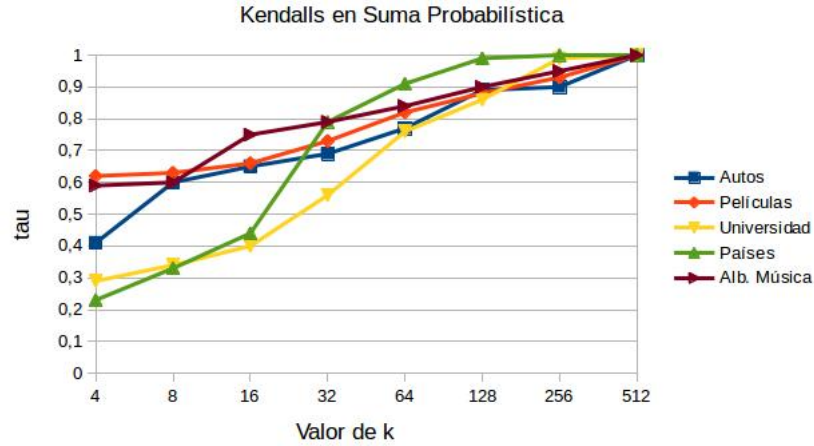


Figura 6.6: Efecto de k en Suma Probabilística

Las pruebas con la tercera y última medida propuesta, Conteo de Intersección arrojaron, los siguientes resultados.

Como se mencionó en la sección anterior, para los dominios de Películas, Álbumes de Música y Modelos de Autos, no se pudo realizar la prueba para $k = 512$, por la cantidad de entidades que estos dominios tienen; en estos caso se realizaron las pruebas con $k = 256$ como el valor más alto de comparación.

En la Figura 6.7 se muestra gráficamente los datos de la Tabla 6.8.

Se puede notar que podemos obtener con $k = 128$ una correlación del 0,8 a los resultados de $k = 512$, lo que a primera impresión no parece ser algo bueno, sin embargo si recordamos el costo computacional de cada una de las medidas ($O(m \log m + nk^2 \log n)$ y $O(m \log m + nk^3 \log n)$), tener una aproximación con una correlación de 0,8 no es tan malo si lo comparamos con el costo computacional que reducimos.

k	Países	Universidades	Películas	Alb. Música	Autos
4	*-0,27	*0,07	*0,34	*0,33	*0,55
8	*0,13	*0,59	*0,76	*0,60	*0,76
16	*0,06	*0,59	*0,78	*0,66	*0,76
32	*0,16	*0,57	*0,83	*0,68	*0,89
64	*0,56	*0,74	*0,89	*0,92	*0,91
128	*0,99	*0,86	*0,88	*0,90	*0,89
256	*0,95	*0,99	*1,00	*1,00	*1,00
512	*1,00	*1,00			

Tabla 6.8: Efecto de k en Conteo de Intersección

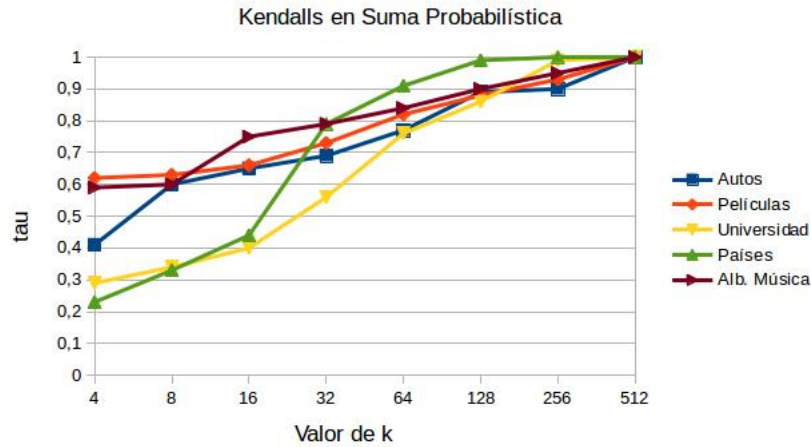


Figura 6.7: Efecto de k en Conteo de Intersección

6.3.4. Análisis del efecto de k

Según lo mostrado en la sección anterior, surgen las siguientes preguntas ¿Un valor alto de k garantiza mejores resultados de las medidas? ¿Qué valor de k se debe utilizar?

Si se analiza la implicancia de k en las medidas, vemos que este valor determina, que no tomaremos en cuenta los vecinos (p, o) que más de k sujetos los tiene en común; dicho de otra manera, ignoramos pares de propiedades y valores que las tienen más de k sujetos.

Entonces, si hay un vecino con más de k sujetos, al no contar ese vecino en particular, afectará el puntaje final de k entidades. Así que variar el valor de k puede afectar los puntajes de muchas entidades. Pero estamos más interesados en los rankings, no los puntajes, y los resultados muestran que los rankings mantienen una fuerte correlación si uno no baja el valor de k demasiado. Dicho eso, en las últimas dos medidas, los vecinos con muchos sujetos tendrán menos influencia en el orden de los resultados.

Sin embargo, sabiendo que las medidas tienen un costo de $(O(m \log m + nk^2 \log n))$ y $O(m \log m + nk^3 \log n)$ y dependiendo el tiempo y recursos que se tienen para obtener los resultados, se puede escoger un valor de k conveniente para los recursos.

6.4. Análisis de la Calidad de los Resultados

Además de analizar el rendimiento de cada uno de las medidas, se desarrollaron pruebas para evaluar ¿Qué medida se aproxima más a la percepción de similitud de las personas? Con esto no sólo se evaluó el tiempo en calcular los resultados, también se analizó si los resultados tiene sentido, al hablar de similitud.

Para esto se hizo uso de dos servicios de recomendación: uno de películas y otro de música. Uno es **BestSimilar** el cual tiene en cuenta diferentes parámetros de películas y utiliza sugerencias y evaluaciones de expertos ("seleccionadas a mano") ¹. En la Figura 6.8 se muestra un ejemplo de recomendación de esta página, donde cada persona puede votar por (Sí/No) si le parece similar este par de películas. Este servicio por cada película provee otras 10 películas similares a la seleccionada.

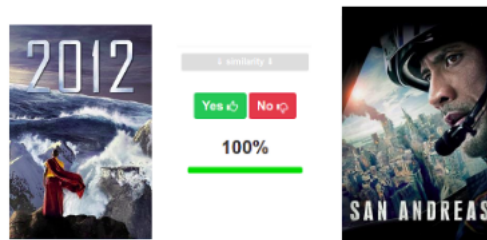


Figura 6.8: Ejemplo de Recomendación BestSimilar

El segundo servicio de recomendación fue **Last.fm**, el cual es muy popular para la recomendación de música ²; aquí por cada objeto (Canción, Artista o Album de música) nos provee 20 objetos similares al seleccionado. La Figura 6.9 muestra un ejemplo de recomendación de este servicio. Aunque no se publica información sobre cómo se calculan los objetos similares, Last.fm cuenta con mucha información (explícita e implícita) sobre la preferencia de más de 60 millones de usuarios.



Figura 6.9: Ejemplo de Recomendación Last.fm

¹<https://bestsimilar.com/>

²<https://www.last.fm/>

Para realizar las pruebas se tomaron los datos indicados en la Tabla 6.9:

Servicio	Entidades	Tamaño del Ranking	Tipo
BestSimilar	100	10	Películas
Last.fm	100	20	Álbums de Música

Tabla 6.9: Datos para prueba de Calidad

Dichos datos fueron seleccionados de forma aleatoria; en la Tabla 6.10 se muestra un ejemplo de los resultados que arrojan estos servicios; se muestran el top 10 de películas similares a *Mad Max: Fury Road (2015)* según BestSimilar ³, y en la segunda columna se muestran los 10 Álbums más similares a *Natural Mystic-Bob Marley* según Last.fm ⁴.

Posición	Mad Max: Fury Road (2015)	Natural Mystic-Bob Marley
1	Mad Max (1979)	Jerusalem -Alpha Blondy
2	Dredd (2012)	True Love- Toots and The Maytals
3	Mad Max 2: The Road Warrior (1981)	The Same Song- Israel Vibration
4	The Book of Eli (2010)	Welcome to Jamrock- Damian Marley
5	Doomsday (2008)	Reggae Greats- Lee "Scratch"Perry
6	Mad Max Beyond Thunderdome (1985)	International Herb - Culture
7	Escape from L.A. (1996)	Promises And Lies- UB40
8	Mortal Engines (2018)	True Democracy- Steel Pulse
9	Waterworld (1995)	The Wailing Wailers- The Wailers
10	Fist of the North Star (1995)	Hebron Gate- Groundation

Tabla 6.10: Ejemplo de Datos

Para realizar las pruebas se calcularon los resultados de cada una de las medidas; luego se encontró el ranking local de cada una de los 100 películas y de los 100 Álbums de Música, seleccionadas en las muestras. Para encontrar el ranking local de un elemento X , simplemente se tomó el ranking global y se seleccionó los resultado $((X, A), s)$ del ranking global y se ordenan los resultados en base a similitud s .

Al tener los rankings locales, se calculó el coeficiente τ_{a-b} de Kendall, entre los resultados de las páginas y los resultados locales de las medidas de cada elemento, y se promediaron los resultados de los 100 elementos de cada dominio.

Los resultados de las pruebas de comparación entra las medidas y el servicio de recomendación de películas BestSimilar se presentan en la Tabla 6.11.

Los resultados anteriores se muestran en el gráfico de la Figura 6.10:

³<https://bestsimilar.com/movies/21434-mad-max-fury-road>

⁴<https://www.last.fm/es/music/Bob+Marley/Natural+Mystic>

Medida	Mínimo	Primer Cuartil	Mediana	Tercer Cuartil	Máximo
Vecinos Comunes	0,3577	0,5018	0,6224	0,7621	0,8558
Suma Probabilística	0,2998	0,4347	0,5485	0,6389	0,7982
Conteo de Intersección	0,7332	0,8022	0,8630	0,9320	0,9827

Tabla 6.11: Resultados de Correlación en BestSimilar

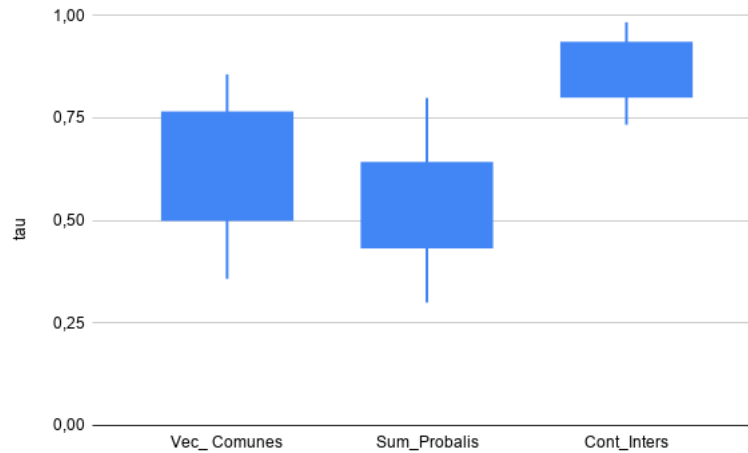


Figura 6.10: Resultados de Correlación en BestSimilar

En cuanto a la comparación de los resultados en Álbumes de Música de Last.fm, se recogieron los resultados presentados en la Tabla 6.12 y mostrados en el gráfico de la Figura 6.11.

Medida	Mínimo	Primer Cuartil	Mediana	Tercer Cuartil	Máximo
Vecinos Comunes	0,1777	0,3402	0,4601	0,6168	0,7549
Suma Probabilística	0,0982	0,2382	0,3836	0,5304	0,6758
Conteo de Intersección	0,4902	0,5483	0,6538	0,7319	0,8377

Tabla 6.12: Resultados de Correlación en Last.fm

Como se puede observar en los dos gráficos, la medida propuesta de Conteo de Intersección, es la que mejor aproxima la percepción de similitud de las personas en los dos dominios, y además de que tiene mejores resultados, estos están menos dispersos; esto se puede explicar ya que la tercera medida considera la dependencia de vecinos.

Considerando el conteo de intersección, aunque las correlaciones medianas en ambos casos 0,8630 y 0,6538 representan correlaciones positivas fuertes y regulares, no reflejan exactamente los resultados de los dos sitios. Sin embargo, si consideramos que estamos acercándonos a una percepción de similitud, sólo tomando los datos (director, actores, etc) que describen una película, es una medida que puede servir de apoyo para un sistema recomendador. Se pueden usar los resultados de la medida Conteo de Intersección para tener unos valores iniciales de similitud entre las entidades sin tener información de los usuarios.

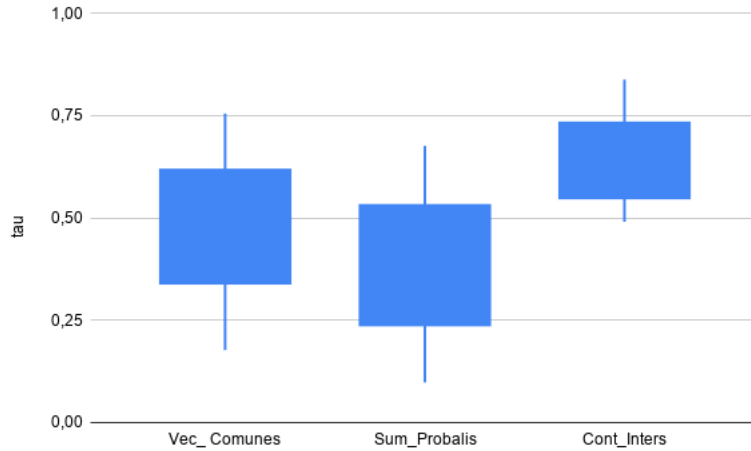


Figura 6.11: Resultados de Correlación en Last.fm

6.5. Comparación de los Resultados con Medida Base

Como se mencionó en el capítulo de trabajos relacionados, una de las medidas de similitud más populares para grafos es SimRank [17], la cual usa el principio de que **dos objetos son similares si están relacionados con objetos similares**.

Para realizar las pruebas se utilizaron varias implementaciones de esta medida de diferentes autores; sin embargo el mayor problema de las implementaciones es la cantidad de memoria que se requiere: debido a que esta medida es recursiva se debe ir guardando todos los resultados, y el peor de los casos es tener un valor real para cada par de nodos del grafo, es decir la medida tiene un costo cuadrático de espacio en memoria.

Al final se usaron dos implementaciones -una en Python ⁵ y otra en Java ⁶-, para comprobar que los resultados fueran iguales, y no existieran errores en las implementaciones; para poder realizar las pruebas se tuvo que reducir la cantidad de datos, por el problema de memoria antes descrito.

Domino	Entidades	Predicados	Objetos	Triples
Películas	22	765	186456	275123
Modelos de Autos	38	89	3115	356416
Países	32	552	66383	74653
Universidades	23	654	106318	74653
Álbum de Música	22	675	100654	169963

Tabla 6.13: Descripción de Datos utilizado para SimRank.

Lo primero que se evaluó fue, ¿Cómo se relacionan los resultados de las medidas propuestas con SimRank? Para ello se encontraron los resultados de SimRank y de las medidas propuestas en los datos descritos en la Tabla 6.13, y se calculó el coeficiente τ -b.

⁵<https://gist.github.com/chao-he/b99912ce50f84b3cda6f>

⁶<http://webla.sourceforge.net/>

Domino	Vecinos Comunes	Sum. Probabilística	Cont. Intersección
Películas	*0,30	*0,32	*0,75
Modelos de Autos	*0,35	*0,46	*0,94
Países	*0,46	*0,56	*0,91
Universidades	*0,56	*0,49	*0,88
Álbum de Música	*0,33	*0,25	*0,89

Tabla 6.14: Comparación de las medidas propuesta con SimRank.

Como se puede observar, la medida que más se acerca a SimRank es Conteo de Intersección; luego surge la pregunta ¿Qué tanto se aproxima SimRank a la percepción de similitud de las personas?. Para intentar responder esta pregunta se realizaron pruebas con los datos reducidos, y compararlo con los dos servicios de recomendación.

Obtenemos los resultados de la Tabla 6.15 en el dominio de Película, comparándolo con el servicio BestSimilar.

Medida	Mínimo	Primer Cuartil	Mediana	Tercer Cuartil	Máximo
Vecinos Comunes	0,0013	0,0743	0,2137	0,3309	0,4689
Suma Probabilística	0,0026	0,0789	0,1416	0,2039	0,3275
Conteo de Intersección	0,2291	0,3290	0,4300	0,4990	0,5789
SimRank	0,2739	0,3738	0,4749	0,5438	0,6238

Tabla 6.15: Resultados de Calidad con Datos Reducidos en Películas

Gráficamente los resultados se ven en la Figura 6.12:

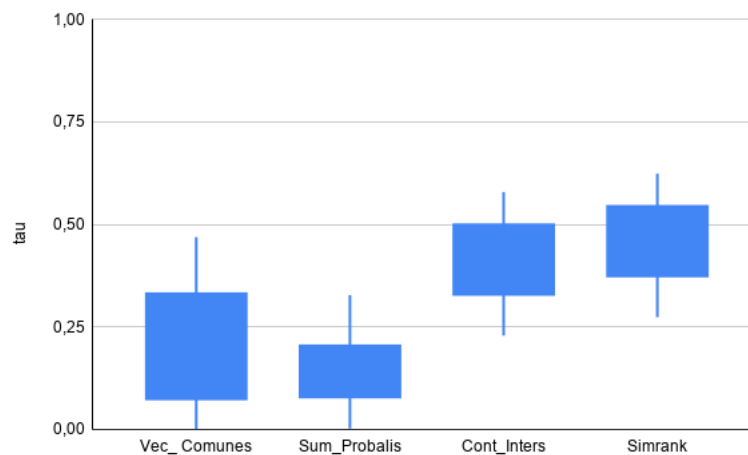


Figura 6.12: Resultados de Correlación con Datos Reducidos en BestSimilar

En cuanto a Álbumes de Música y los resultados de Last.fm se obtuvieron los resultados en la Tabla 6.16.

Medida	Mínimo	Primer Cuartil	Mediana	Tercer Cuartil	Máximo
Vecinos Comunes	0,0028	0,0869	0,1482	0,3132	0,4087
Suma Probabilística	0,0008	0,0685	0,1441	0,2097	0,2883
Conteo de Intersección	0,1763	0,2620	0,3274	0,4330	0,5234
SimRank	0,2317	0,3174	0,3829	0,4884	0,5789

Tabla 6.16: Resultados de Correlación con Datos Reducidos en Álbumes de Música

En la Figura 6.13 se muestran los resultados.

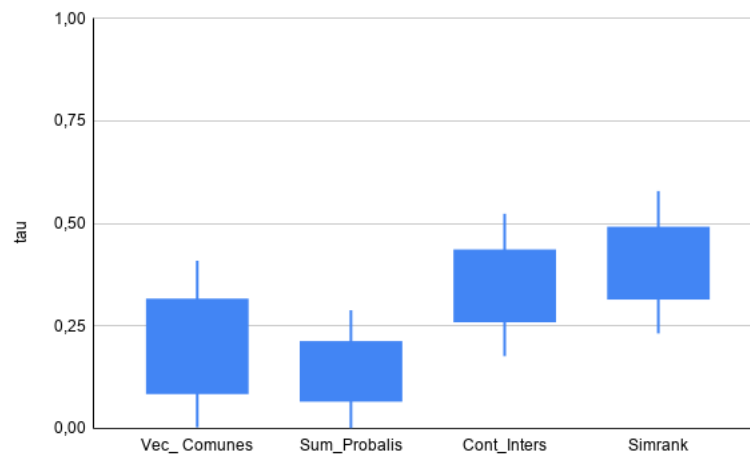


Figura 6.13: Resultados de Correlación con Datos Reducidos en Last.Fm

Debemos recordar que las medidas fueron aplicadas a un conjunto reducido de datos; esto podría explicar la reducción en la calidad de los resultados.

Otra cosa a notar en los dos gráficos es que SimRank supera a las medidas propuestas en cuanto a la aproximación de similitud de las personas; sin embargo ahora se mostrará (Tabla 6.17) los tiempos en que se realizaron dichas pruebas, para comparar el rendimiento de las medidas.

Medida	Mínimo	Primer Cuartil	Mediana	Tercer Cuartil	Máximo
Vecinos Comunes	243,37	256,24	270,33	284,98	300,88
Suma Probabilística	320,62	333,49	348,43	362,22	378,123
Conteo de Intersección	643,39	650,50	660,11	669,3008	756,125
SimRank	1148,89	1190,30	1227,25	1269,15	1305,65

Tabla 6.17: Tiempo en segundos de Pruebas de Correlación con Datos Reducidos

En la Figura 6.14 se muestra una gráfica con los tiempos

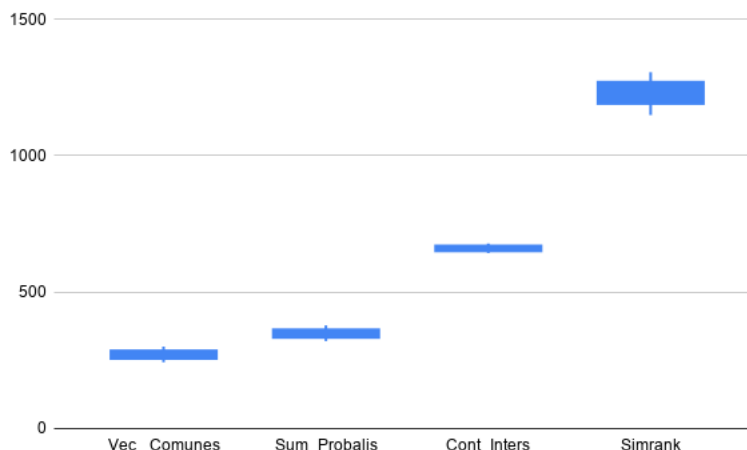


Figura 6.14: Tiempo en segundos de Pruebas con Datos Reducidos

Los datos de los tiempos de la Tabla 6.17 son de las pruebas en los dos servicios, tanto en BestSimilar como en Last.fm; esto debido a que los tiempos registrados están en los mismo rangos, por que la cantidad de entidades son iguales en ambos dominios, como se puede ver en la Tabla 6.13.

6.6. Aplicación de las medidas en aristas inversas

En las pruebas anteriores, se seleccionaron las triples (s, p, o) donde el sujeto s pertenece a un dominio específico; dicho de otra manera, se toman las aristas que salen de los sujetos. En este capítulo se estudiará el efecto que tiene en los resultados, incluir las aristas (s, p, o) donde el objeto o es una entidad del dominio; es decir incluir las aristas que llegan a los sujetos de sus vecinos.

Para extraer estas tuplas, primero se identifican las triples (s, p, o) donde o es una entidad del dominio; luego se guardan estos triples pero en el orden inverso (o, p, s) , para que las implementaciones de las medidas de similitud pueda ser aplicadas sin mayor problema, recordando que lo importante para estas pruebas es medir cómo varían los resultados con la inclusión de esta información.

En la Tabla 6.18 se muestra cómo cambian los conjuntos de datos al incluir estas nuevas triples.

De igual manera se realizó esta prueba con SimRank, pero para esta medida se usaron los datos mostrados en la Tabla 6.19:

Es claro pensar que el número de predicados, objetos y triples van a aumentar, ya que estamos agregando aristas que antes no consideramos; pero el número de entidades es constante, ya que no estamos agregando nuevas entidades del dominio; todas fueron seleccionadas

Domino	Entidades	Predicados	Objetos	Triples
Películas	765	1265	296396	857895
Modelos de Autos	38	195	15652	758826
Países	179	1456	256956	512365
Universidades	842	19556	136459	215652
Álbum de Música	653	2564	315654	354654

Tabla 6.18: Descripción de dominios con arista en dos direcciones.

Domino	Entidades	Predicados	Objetos	Triples
Películas	22	1650	232520	313156
Modelos de Autos	38	195	15652	758826
Países	32	856	95365	123565
Universidades	23	1356	185653	145236
Álbum de Música	22	1563	195653	352632

Tabla 6.19: Descripción de Datos utilizado para SimRank con aristas en dos direcciones.

en las primeras pruebas.

Para realizar las pruebas, las medidas fueron aplicadas a este nuevo conjunto de datos, con un valor de $k = 512$ para las medidas de Conteo de Vecinos Comunes y Suma Probabilística, y para Conteo de Intersección se usó $k = 256$; luego se calculó el coeficiente de correlación τ -b comparando estos nuevos resultados con los resultados de las pruebas anteriores (Tabla 6.1 y Tabla 6.13).

Los resultados de estas pruebas se muestran en la Tabla 6.20.

Domino	Vec. Comunes	Sum. Proba	Conte. Inter	SimRank
Películas	*0,658	*0,565	*0,756	*0,736
Modelos de Autos	*0,789	*0,654	*0,827	*0,789
Países	*0,726	*0,685	*0,865	*0,823
Universidades	*0,645	*0,665	*0,785	*0,806
Álbum de Música	*0,546	*0,516	*0,654	*0,654

Tabla 6.20: Correlación de Resultados con Aristas en dos Direcciones.

La Figura 6.15 muestra de manera gráfica los resultados.

Seguidamente se realizaron pruebas para medir el tiempo que toma calcular las medidas respecto al valor de k en los dominios de Películas y Álbumes de música, igual que las mostradas en la sección 6.2.

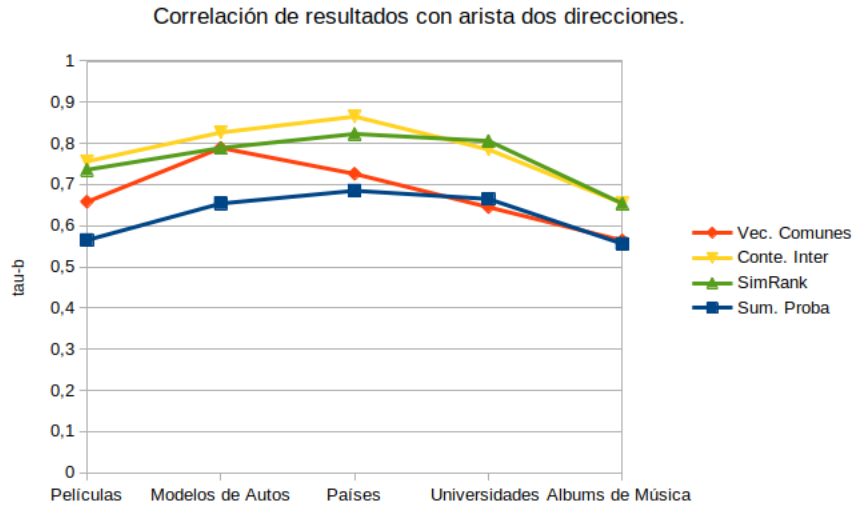


Figura 6.15: Correlación de Resultados con Aristas en dos Direcciones.

La Tabla 6.21 muestra los tiempos en segundos que se necesitaron para calcular los resultados:

Valor de k	Vec. Comunes	Sum. Proba	Conte. Inter
4	558	590	1170
8	533	557	1246
16	533	612	1472
32	580	633	2725
64	721	729	7932
128	796	809	12313
256	897	889	21490
512	1609	1167	

Tabla 6.21: Tiempo en Segundos para Películas con Aristas en dos Direcciones

Si se comparan los resultados de la Tabla 6.2 con los tiempos en de la Tabla 6.21, vemos que se aumentó el tiempo en calcular los resultados; esto se debe a que hay que procesar más triples.

En la Tabla 6.22 se muestran los tiempos en Álbumes de Música para calcular las medidas; si se comparan con los tiempos de la Tabla 6.5 vemos que aumenta el tiempo en procesar los datos.

Por último se comparó si al agregar esta información aumentaba la correlación de los resultados, por lo que se realizaron pruebas iguales a las de la sección 6.5 con los datos de la Tabla 6.19.

Valor de k	Vec. Comunes	Sum. Proba	Conte. Inter
4	525	611	1159
8	602	571	1091
16	574	664	1683
32	557	695	2506
64	688	728	7994
128	819	810	13004
256	965	941	18728
512	1941	1222	

Tabla 6.22: Tiempo en Segundos para Álbumes de Música con Aristas en dos Direcciones

Medida	Mínimo	Primer Cuartil	Mediana	Tercer Cuartil	Máximo
Vecinos Comunes	0,0039	0,0866	0,1865	0,2260	0,5239
Suma Probabilística	0,0039	0,0867	0,1865	0,2265	0,3854
Conteo de Intersección	0,2345	0,3245	0,4245	0,4854	0,5234
SimRank	0,2895	0,3654	0,4654	0,5568	0,6566

Tabla 6.23: Resultados de Correlación con Datos Reducidos en Películas con Aristas en dos Direcciones

El gráfico de la Figura 6.16 muestra los resultados:

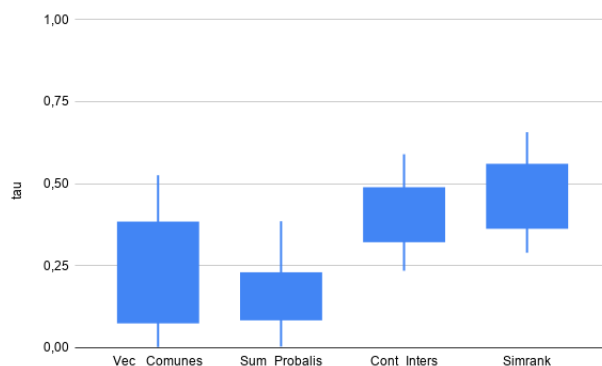


Figura 6.16: Resultados de Calidad en BestSimiliar con Aristas en Dos Direcciones.

Además se comparó la calidad de los resultados en Álbumes de música usando aristas en dos direcciones, obteniendo los siguientes resultados:

La Figura 6.17, muestra gráficamente los resultados de la Tabla 6.24:

Si se comparan las Figuras 6.16 y 6.12, y comparamos las Figuras 6.13 y 6.17, vemos que agregar esta información mejora un poco los resultados, pero además aumenta el tiempo que se necesita para calcular los resultados. Por lo que considerando esta información, aumentará el tiempo no así la calidad de los resultados sustancialmente.

Medida	Mínimo	Primer Cuartil	Mediana	Tercer Cuartil	Máximo
Vecinos Comunes	0,0027	0,0895	0,1524	0,3865	0,4265
Suma Probabilística	0,0089	0,0757	0,1546	0,2565	0,3565
Conteo de Intersección	0,1856	0,2865	0,3569	0,4256	0,5854
SimRank	0,2655	0,3854	0,4569	0,5245	0,6254

Tabla 6.24: Resultados de Correlación con Datos Reducidos en Álbumes de Música con Aristas en dos Direcciones

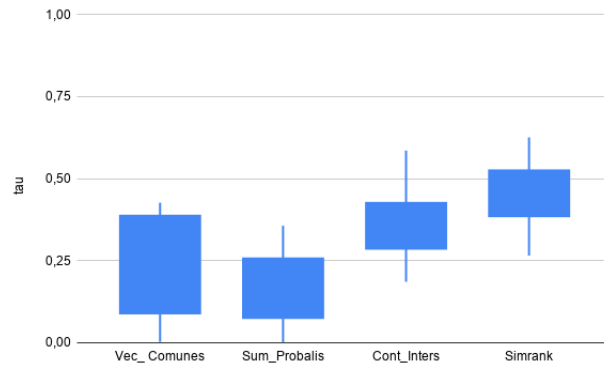


Figura 6.17: Resultados de Correlación en Last.fm con Aristas en dos Direcciones.

Conclusión

Uno de los objetivos de esta investigación era identificar medidas de similitud existentes y ver cómo se pueden aplicar estas medidas en Wikidata. SimRank es una medida global de similitud que se aplica a grafos dirigidos pero no etiquetados; se logró aplicar a una porción de los datos ajustando las tuplas (s, p, o) a los pares $(s, (p, o))$ y con esto volver un grafo etiquetado a no etiquetado, sin perder la información de los predicados.

Al trabajar con varias implementaciones de SimRank, se logró identificar que esta medida requiere mucha memoria, ya que la medida requiere almacenar la matriz de adyacencia y a su vez una matriz del mismo tamaño, con la medida de similitud asignada en cada iteración.

En nuestro estudio proponemos 3 medidas de similitud que lograron mejorar el rendimiento y bajar los requisitos de memoria en comparación con SimRank; tanto así que se logró aplicar estas medidas a un conjunto de datos de mayor tamaño, que el que soportó SimRank.

Cada una de las medidas tiene sus fortalezas; sin embargo, en calidad de resultados, la medida de Conteo de Intersección fue la mejor respecto a las medidas propuestas, ya que esta fue pensada intentando superar las debilidades de las otras medidas propuestas.

SimRank tiene un costo de $O(jkn^4)$ en tiempo y $O(n^2)$ en memoria donde n es el número de entidades y j es el número de iteraciones. Las medidas propuestas en este trabajo usan un valor de aproximación k que se puede ajustar según los recursos que se posean para calcular las medidas; con esto se logra que el costo se acote a $O(m \log m + nk^2 \log n)$, donde m es el número de aristas del grafo, para los algoritmos propuestos las medidas de Conteo de Vecinos Comunes y Suma Probabilística.

En cuanto al algoritmo propuesto Conteo de Intersección, tiene un costo de $O(m \log m + nk^3 \log n)$, el cual es mayor a las otras dos medidas propuestas, sin embargo es mejor en tiempo que SimRank, además esta medida logró aproximarse mejor a la percepción de similitud de las personas, y así, a su vez, si fijamos el valor de k a una constante, el costo es igual a los otros dos algoritmos: $O(m \log m)$. Con este valor de aproximación, se logró trabajar en un conjunto de datos muchos más grande, que el conjunto de datos con que se logró aplicar SimRank, y, sobre grafos más pequeños, tenía resultados competitivos con SimRank considerando las opiniones de los usuarios. Además, puede evitar el "doble recuento" de información redundante al considerar un vecindario común como un evento único, en lugar de suponer que consta de múltiples eventos independientes (es decir, vecinos individuales).

La medida de Vecinos Comunes presenta falencias en la calidad de los resultados; esto

debido a que genera muchos empates, pero es una medida que se debe tener en consideración por lo sencilla y práctica de implementar.

La media de Suma Probabilística es afectada directamente por la dependencia entre pares (p, o) sin embargo si se tiene un conjunto de datos RDF, donde no existan este tipo de dependencias, esta medida puede ser aplicada sin problema.

Para trabajos futuros se debe tener en cuenta que hacer uso de un framework como Spark, ayuda en gran medida para el manejo de memoria; otra cosa que se puede rescatar de este trabajo es el uso de las etiquetas de las aristas, considerando los pares (p, o) como vecinos; con este ajuste es posible aplicar medidas de similitud que fueron diseñados para grafos no etiquetados. En el futuro, sería interesante estudiar si se pueden calcular estas medidas de similitud sobre grafos con millones de nodos (como el grafo completo de Wikidata). Además, sería interesante comparar los resultados de los algoritmos sobre otros grafos y dominios.

Bibliografía

- [1] David R Anderson, Dennis J Sweeney, Thomas A Williams, María del Carmen Hano Roa, and Teresa López Álvarez. *Estadística para administración y economía*. International Thomson, 2001.
- [2] Samur Araujo, Jan Hidders, Daniel Schwabe, and Arjen P De Vries. Serimi-resource description similarity, RDF instance matching and interlinking. *arXiv preprint arXiv:1107.1104*, 2011.
- [3] Solomon Atnafu, Lionel Brunie, and Harald Kosch. Similarity-based operators and query optimization for multimedia database systems. In *Database Engineering and Applications, 2001 International Symposium on.*, pages 346–355. IEEE, 2001.
- [4] Phoebe Ayers, Charles Matthews, and Ben Yates. *How Wikipedia works: And how you can be a part of it*. No Starch Press, 2008.
- [5] Vevake Balaraman, Simon Razniewski, and Werner Nutt. Recoin: Relative Completeness in Wikidata. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pages 1787–1792. International World Wide Web Conferences Steering Committee, 2018.
- [6] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284(5):34–43, 2001.
- [7] Dan Brickley, Ramanathan V Guha, and Brian McBride. RDF Schema 1.1. W3C Recommendation. *World Wide Web Consortium, February*, 2014.
- [8] Pável Calado, Marco Cristo, Marcos André Gonçalves, Edleno S de Moura, Berthier Ribeiro-Neto, and Nivio Ziviani. Link-based similarity measures for the classification of Web documents. *Journal of the American Society for Information Science and Technology*, 57(2):208–221, 2006.
- [9] Pavel Chebotarev and Elena Shamis. The matrix-forest theorem and measuring relations in small social groups. *arXiv preprint math/0602070*, 2006.
- [10] Richard Cyganiak, David Wood, and Markus Lanthaler. RDF 1.1 Concepts and Abstract Syntax. World Wide Web Consortium Recommendation, 2014.
- [11] Stefan Decker, Sergey Melnik, Frank Van Harmelen, Dieter Fensel, Michel Klein, Jeen

- Broekstra, Michael Erdmann, and Ian Horrocks. The semantic web: The roles of XML and RDF. *IEEE Internet computing*, 4(5):63–73, 2000.
- [12] Dániel Fogaras and Balázs Rácz. Scaling link-based similarity search. In *Proceedings of the 14th International Conference on World Wide Web*, pages 641–650. ACM, 2005.
- [13] Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on knowledge and data engineering*, 19(3):355–369, 2007.
- [14] Jan Grant and Dave Beckett. RDF test cases. *W3C recommendation*, 10, 2004.
- [15] Bernardo Cuenca Graua, Ian Horrocksa, Boris Motika, Bijan Parsiab, Peter Patel-Schneiderc, and Ulrike Sattlerb. OWL 2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6:309–322, 2008.
- [16] Ioana Hulpuş, Narumol Prangnawarat, and Conor Hayes. Path-based semantic relatedness on linked data and its use to word and entity disambiguation. In *International Semantic Web Conference*, pages 442–457. Springer, 2015.
- [17] Glen Jeh and Jennifer Widom. SimRank: a measure of structural-context similarity. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 538–543. ACM, 2002.
- [18] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.
- [19] Elizabeth A Leicht, Petter Holme, and Mark EJ Newman. Vertex similarity in networks. *Physical Review E*, 73(2):026120, 2006.
- [20] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.
- [21] Michael Luggen, Djellel Difallah, Cristina Sarasua, Gianluca Demartini, and Philippe Cudré-Mauroux. Non-Parametric Class Completeness Estimators for Collaborative Knowledge Graphs—The Case of Wikidata. In *International Semantic Web Conference*, pages 453–469. Springer, 2019.
- [22] Pedro Morales and Luis Rodríguez. Aplicación de los coeficientes correlación de Kendall y Spearman. *Barquisimeto, Venezuela: Universidad Centroccidental Lisandro Alvarado (UCLA)*, 2016.
- [23] María Dolores Frías Navarro, Juan Pascual Llobell, and José Fernando García Pérez. Tamaño del efecto del tratamiento y significación estadística. *Psicothema*, 12(Su2):236–240, 2000.
- [24] Mark EJ Newman. Clustering and preferential attachment in growing networks. *Physical review E*, 64(2):025102, 2001.

- [25] Phuong Nguyen, Paolo Tomeo, Tommaso Di Noia, and Eugenio Di Sciascio. An evaluation of SimRank and Personalized PageRank to build a recommender system for the Web of Data. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1477–1482, 2015.
- [26] Alina Petrova, Evgeny Sherkhonov, Bernardo Cuenca Grau, and Ian Horrocks. Entity comparison in RDF graphs. In *International Semantic Web Conference*, pages 526–541. Springer, 2017.
- [27] Tomás Saorín and Juan-Antonio Pastor-Sánchez. Enriquecimiento de entidades de Wikidata mediante un modelo de descomposición y mapeado de categorías de Wikipedia. In *Actas del IV Congreso ISKO España-Portugal 2019, XIV Congreso ISKO España*, 2019.
- [28] Nigel Shadbolt, Tim Berners-Lee, and Wendy Hall. The semantic web revisited. *IEEE intelligent systems*, 21(3):96–101, 2006.
- [29] Abdul Ghaffar Shoro and Tariq Rahim Soomro. Big Data Analysis: Apache Spark perspective. *Global Journal of Computer Science and Technology*, 2015.
- [30] Sidney Siegel, N John Castellan, et al. *Estadística no paramétrica: aplicada a las ciencias de la conducta*, volume 4. Trillas México, 1995.
- [31] Ravi Sinha and Rada Mihalcea. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In *Semantic Computing, 2007. ICSC 2007. International Conference on*, pages 363–369. IEEE, 2007.
- [32] Tomáš Skopal and Benjamin Bustos. On nonmetric similarity search problems in complex domains. *ACM Computing Surveys (CSUR)*, 43(4):1–50, 2011.
- [33] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Random walk with restart: fast solutions and applications. *Knowledge and Information Systems*, 14(3):327–346, 2008.
- [34] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.
- [35] Brit Youngmann, Tova Milo, and Amit Somech. Boosting SimRank with Semantics. In *Advances in Database Technology - 22nd International Conference on Extending Database Technology, EDBT 2019, Lisbon, Portugal, March 26-29, 2019*, 2019.
- [36] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pages 15–28, 2012.
- [37] Matei Zaharia, Reynold S Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J Franklin, et al. Apache Spark: a unified engine for big data processing. *Communications of the*

ACM, 59(11):56–65, 2016.

- [38] Zhipeng Zhang, Yingxia Shao, Bin Cui, and Ce Zhang. An experimental evaluation of SimRank-based similarity search algorithms. *Proceedings of the VLDB Endowment*, 10(5):601–612, 2017.
- [39] Weiguo Zheng, Lei Zou, Wei Peng, Xifeng Yan, Shaoxu Song, and Dongyan Zhao. Semantic SPARQL similarity search over RDF knowledge graphs. *Proceedings of the VLDB Endowment*, 9(11):840–851, 2016.