Procesamiento de lenguaje natural y métodos basados en grafos

Mireya Tovar Vidal
Guillermo De Ita Luna
Pedro Bello López
Meliza Contreras Gónzalez
Fernando Zacarias Flores
Yolanda Moyao Martínez
Luis Carlos Altamirano Robles

Editores

Dirección General de Publicaciones

Procesamiento de lenguaje natural y métodos basados en grafos

Procesamiento de lenguaje natural y métodos basados en grafos

Mireya Tovar Vidal
Guillermo De Ita Luna
Pedro Bello López
Meliza Contreras Gónzalez
Fernando Zacarias Flores
Yolanda Moyao Martínez
Luis Carlos Altamirano Robles
Coordinadores

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
2022

Primera Edición **2022**
ISBN BUAP: 978-607-525-845-4

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA • *Rectora:* Ma. Lilia Cedillo Ramírez • *Secretario General:* José Manuel Alonso Orozco • *Vicerrector de Extensión y Difusión de la Cultura:* Flavio Guzmán Sánchez • *Director General de Publicaciones:* Luis Antonio Lucio Venegas • *Directora de la Facultad de Ciencias de la Computación:* María del Consuelo Molina García

Hecho en México
Made in Mexico

Prólogo

En el presente libro titulado "Procesamiento de lenguaje natural y métodos basados en grafos" se muestran diferentes áreas de investigación en procesamiento del lenguaje natural y métodos que utilizan grafos para su solución que son abordadas por distintos grupos de investigación a nivel nacional.

La obra incluye nueve capítulos de investigación, en las áreas de procesamiento de lenguaje natural, aplicaciones en grafos, web semántica entre otras.

Los capítulos que forman parte de esta obra fueron revisados mediante el sistema de doble par ciego y aprobados para su publicación por expertos en el área de conocimiento, lo que permitió asegurar su calidad científica en las áreas de estudio. A continuación se menciona la aportación de cada uno de ellos.

En el Capítulo 1 se presenta un método para la generación de n-gramas usados en la extracción de frases clave en documentos científicos y la extracción de relaciones semánticas de tipo sinónimo e hipónimo a través de patrones léxico-sintácticos. En el Capítulo 2 se presenta un método para evaluar medidas de similitud semántica, el cual determina la medida que arroja mejores resultados, con respecto a ciertos criterios de evaluación. En el Capítulo 3 se revisa la implementación de dos modelos preentrenados basados en BERT para el análisis de polaridad de tuits (tweets) en español de diferentes países hispanohablantes. En el Capítulo 4 se presenta un enfoque para el descubrimiento de tópicos a partir de artículos científicos en español sobre el dominio de salud, utilizando el algoritmo LDA (de su nombre en inglés *Latent Dirichlet Allocation*). En el Capítulo 5 se hace una revisión de los principales métodos de comparación existentes en la literatura para evaluar el grado de similitud entre dos cadenas de texto. En el Capítulo 6 se hace uso de ontologías para modelar el conocimiento operativo y de dominio de los repositorios institucionales de acuerdo con los lineamientos generales y técnicos del CONACYT. Además, se presenta una interfaz accesible vía web, para realizar consultas tanto para usuarios expertos como los no expertos en tecnologías semánticas. En el Capítulo 7 se realiza una revisión y análisis de diferentes métricas arbóreas para grafos, que pueden generalizarse a hipergrafos, de tal forma que con la métrica y parámetro adecuados se puedan encontrar soluciones parámetro fijo tratable para el análisis de gramáticas de reemplazo de hiperaristas. En el Capítulo 8 se presenta un método novedoso para el conteo de conjuntos independientes sobre estructuras tipo malla. Se parte de explicar las recurrencias que usa el método para contar conjuntos independientes sobre topologías básicas de grafos. Por último, en el Capítulo 9 se muestra cómo las propiedades de la secuencia $\beta_{i,j}$ que representa el producto entre dos números Fibonacci $F_i F_j$ se puede utilizar para el cálculo del índice de Merrifield-Simmons en grafos bipoligonales.

Finalmente, queremos agradecer a cada uno de los autores por su aportación, a nuestros revisores por su valiosa labor y cuidado en el proceso de revisión, a la Facultad de Ciencias de la Computación de la Benemérita Universidad

Autónoma de Puebla y a todos aquellos cuya participación contribuyó para la publicación de este libro.

Los editores,
Mireya Tovar Vidal
Guillermo De Ita Luna
Pedro Bello López
Meliza Contreras Gónzalez
Fernando Zacarias Flores
Yolanda Moyao Martínez
Luis Carlos Altamirano Robles

Índice general

Capítulo 1

Una aproximación basada en n-gramas para la detección de frases clave y relaciones semánticas

Ana Laura Lezama Sánchez[1], Mireya Tovar Vidal[1], José A. Reyes-Ortiz[2]
[1] Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
[2] Universidad Autónoma Metropolitana, Azcapotzalco

ana.lezama@alumno.buap.mx, mireya.tovar@correo.buap.mx,
jaro@azc.uam.mx

Resumen. Las frases clave tienen por objetivo encapsular la idea principal de un fragmento de texto proporcionando una percepción resumida del contenido. Sin ellas no se podría entender la idea general. Suelen utilizarse para algunas tareas, por ejemplo, la generación automática de resúmenes. Por otro lado, las relaciones semánticas son segmentos que pueden ser usados en varias tareas como la formación de conceptos, jerarquías y la existencia de relaciones no jerárquicas; además están relacionadas de acuerdo con su significado. En el presente trabajo se presenta la extracción de frases clave generando *n*-gramas y posteriormente la extracción de relaciones de tipo sinónimo e hipónimo con patrones léxico sintácticos de la literatura.

Palabras Clave: relaciones semánticas, frases clave, *n*-gramas, patrones

1 Introducción

Las frases clave son las que capturan la idea principal en un texto. Para el lector son fundamentales para obtener la idea general por lo que son fáciles de identificar y extraer para un posterior uso. Sin embargo, cuando se desean extraer de manera automática se debe de recurrir a procedimientos computacionales (métodos) que sean capaces de identificarlas en poco tiempo y de manera precisa (Platero 2019). Por lo tanto, la extracción automática de frases clave reduce factores como el tiempo de respuesta, es decir, el lector o usuario no necesita leer los textos, bastará con proporcionarlos al método de extracción de frases clave y este le proporcionará la información necesaria. Algunas de ellas comparten información que las relaciona y generan una idea completa, es decir pueden presentar relaciones de tipo sinonimia o hiponimia que no alteran el significado del texto, sino que lo describen de diferente manera (Platero 2019). Las relaciones semánticas son parte fundamental en el lenguaje humano. Ellas dan forma a una idea en una oración. Algunos tipos de relaciones semánticas son de tipo sinonimia, hiponimia e hiperonimia, meronimia, holonimia entre otros. En este trabajo solo se abordaron relaciones de tipo sinonimia e hiponimia (Aggarwal, 2018). Las relaciones semánticas de tipo hiponimia, son aquellas donde existe una relación

que incluye semántica de un término en otro. Las relaciones de tipo sinonimia son aquellas donde existe una relación entre dos o más palabras que tienen el mismo significado y pertenecen a la misma parte del discurso, pero se escriben de manera diferente (Akhtyamova et al., 2017). En el ámbito computacional la extracción de relaciones semánticas es una tarea importante para un método o algoritmo en específico. El área de estudio que se encarga de generar modelos computacionales para llevar a cabo la extracción de relaciones semánticas y frases clave es el Procesamiento del Lenguaje Natural (*PLN*). Es el encargado de modelar el conocimiento para que sea procesado "entendido" por una computadora y obtenga el conocimiento deseado en poco tiempo y con la calidad deseable. El concepto de n-grama o el pesado *tf-idf* son utilizados en algunas investigaciones durante el desarrollo de modelos para el tratamiento de texto. Los n-gramas son las secuencias de palabras que aparecen en los textos (Vuotto et al., 2015). Por otro lado, el pesado *tf-idf* genera listas de palabras con un peso que indica qué tan relevante es la palabra con respecto a un documento seleccionado (Sidorov 2013). Por ejemplo, el acrónimo *XAS* es el significado de *X-ray absorption spectroscopy* y se trata de una relación de sinonimia y ambas son consideradas frases clave en los documentos de entrenamiento en *SemEval* 2017 tarea 10 subtareas 1 y 3.

SemEval es una competencia dedicada a la evaluación de sistemas computacionales para el análisis semántico con problemas del área del Procesamiento del Lenguaje Natural (*PLN*) y sus áreas de estudio. En el año 2017 propusieron 12 tareas diferentes. En este trabajo se centra en la tarea 10, es decir subtareas: 1 para la extracción automática de frases clave y 3 para la extracción de relaciones semánticas de tipo sinonimia e hiponimia en las frases clave previamente identificadas (Albawi et al., 2017).

Este trabajo propone llevar a cabo la tarea de extraer frases clave por medio de n-gramas y un pesado *tf-idf* en textos sobre publicaciones científicas. Posteriormente extraer las relaciones semánticas de tipo sinonimia e hiponimia presentes en el corpus con patrones léxico sintácticos. Además, se realizó una búsqueda de las frases clave en los repositorios de relaciones y se extrajo el tipo de relación entre ellas.

El documento se encuentra dividido en 5 secciones. La primera sección presenta un estudio del estado del arte en materia de extracción de relaciones semánticas y frases clave. Por otro lado, la sección 2 expone el método propuesto para la extracción de las frases clave y para las relaciones semánticas. La sección 3 muestra el conjunto de datos utilizados durante los experimentos. Luego la sección 4 expone los resultados obtenidos al aplicar el método propuesto. La sección 5 detalla las conclusiones obtenidas y finalmente las referencias utilizadas en este trabajo.

2 Estado del arte

En este apartado se presentan trabajos relacionados con la tarea de extracción de relaciones semánticas por medio de métodos tradicionales como patrones léxico sintácticos, diccionarios, entre otros.

Un método de minería de relaciones semánticas entre genes, trastornos y fármacos provenientes de diferentes conjuntos de datos biomédicos es propuesto por (Al-Zaidy et al., 2018). Los autores usan la enfermedad de Parkinson como un caso de estudio y se enfocan en extraer las relaciones entre el trastorno, gen y el fármaco de la enfermedad a partir de cuatro conjuntos de datos biomédicos. Los conjuntos de datos utilizados en las pruebas realizadas fue *SemMedDB, KEGG, Uniprot* y *PharmGKB*, todo ellos contienen patrones de dominio para trastornos, sustancias químicas y fármacos, genes y secuencias moleculares. Cada conjunto de datos fue convertido en formato *RDF* y posteriormente propusieron un algoritmo para extraer relaciones semánticas de cada conjunto de datos.

En (Bentrcia et al., 2018) presentan un método para el desarrollo de gramáticas con el objetivo de extraer algunas relaciones semánticas usadas en el campo genérico-específico, parte-todo, ubicación, causa y función. El objetivo de los autores es proporcionar un método accesible y de fácil uso para encontrar contextos ricos en conocimiento. El método propuesto busca cada relación en la que participa cada palabra. Los autores generaron "bocetos" de palabras que representan diferentes relaciones como verbo-objeto, modificadores o frases preposicionales. Los resultados experimentales fueron hechos con el corpus *EcoLexicon* en inglés de dominio medioambiental. Los autores generan una formalización de patrones gramaticales en forma de expresiones regulares combinadas con etiquetas *PoS*. Además, consideraron la relación semántica de tipo hiponimia. En total generaron 56 diferentes gramáticas considerando diferentes aspectos específicos en cada relación. El método fue evaluado por medio de las métricas precisión y exhaustividad para evaluar la calidad de las gramáticas generadas.

En (León-Araúz et al., 2016) proponen un método híbrido que tiene como objetivo enriquecer la construcción automática de una ontología del Corán. Los autores explotaron los patrones conjuntivos árabes que existen en la gramática árabe tradicional. El método extrae frases conjuntivas y relaciones semánticas apoyándose de la regla de la conjunción propia de la gramática árabe. Los autores utilizan un conjunto de patrones para la extracción de frases conjuntivas. Además, llevaron a cabo una combinación de pruebas estadísticas y los resultados obtenidos por expertos en el dominio. También de manera manual obtienen tres categorías diferentes de relaciones semánticas de todo el Corán que son antonimia, género y clase. Posteriormente extraen un conjunto de palabras que forman términos de una ontología. Los autores utilizan un método de filtrado basado en un coeficiente de correlación para seleccionar relaciones sólidas. El método propuesto cuenta con una fase de extracción de términos que incluyó la extracción de sustantivos, nombres propios y adjetivos en su forma raíz. Las métricas de evaluación utilizadas fueron precisión y exhaustividad obteniendo un 84% y 92% respectivamente.

Un método híbrido que combina el Procesamiento del Lenguaje Natural y técnicas estadísticas para la extracción de relaciones semánticas de documentos extraídos de la librería digital *ACM* con el objetivo de enriquecer una ontología de dominio es expuesta por (Shanidze et al., 2019). Las relaciones semánticas extraídas fueron sinónimos,

hipónimos, hiperónimos, parte de, hecho de, atributo de, delimitado por, tiene lugar en, resultado de, afecta, etc. La base de datos léxica *WordNet* es utilizada para la construcción de relaciones de tipo sinónimos, hipónimos e hiperónimos. Para la construcción de las demás relaciones proponen un algoritmo diferente que refina la oración, es decir, elimina palabras innecesarias en la oración en función de un árbol de dependencias generado. La evaluación es llevada a cabo por las métricas de precisión, exhaustividad y medida-F_1.

En (Ta et al., 2016) exponen un algoritmo para la extracción de relaciones semánticas con un enfoque basado en reglas. Los autores sugieren identificar verbos entre un sujeto y un objeto para obtener una secuencia de relaciones semánticas de Wikipedia. Además, emplean *synsets* presentes en *WordNet* para la extracción de relaciones semánticas entre conceptos y sus sinónimos del corpus de texto. El algoritmo procesa 200 artículos de Wikipedia del dominio de tecnologías de la información. La búsqueda se amplía a sujetos y objetos de los predicados y de esta manera obtienen los sinónimos desde *WordNet* de los sujetos y objetos identificados previamente.

En (Zhang et al., 2020) proponen un sistema para la extracción de relaciones semánticas entre entidades en artículos académicos haciendo uso de patrones sintácticos extraídos de la literatura de tipo hipónimo-hiperónimo centrándose en el resumen y palabras clave de los documentos a analizar. El sistema extrae entidades semánticas como conceptos e instancias con sus atributos del texto completo. Las entidades extraídas fueron sustantivos y frases nominales. Las fuentes de datos externas de la web, como el grafo de conceptos de *Microsoft* fueron usados para la evaluación de la calidad de los conceptos y relaciones extraídas. Los conceptos fueron usados para construir una taxonomía científica que cubra el contenido de investigación de los documentos. En la evaluación del sistema aplicaron su enfoque en un conjunto de diez mil documentos académicos y llevaron a cabo varias evaluaciones para demostrar la efectividad del sistema propuesto. Las métricas utilizadas en la evaluación del sistema fueron precisión y exhaustividad.

3 Método propuesto

En esta sección se describe el método propuesto para la extracción de frases clave y relaciones semánticas para dar solución a las tareas 1 y 3 de *SemEval* 2017 tarea 10 (Augenstein et al., 2017). Primero se extrajeron las frases clave basándose en la extracción de *n*-gramas y un pesado *tf-idf* de términos que aparezcan en el 50% de los documentos. Posteriormente se creó un repositorio de patrones léxico sintácticos para relaciones de tipo sinonimia, hiponimia e hiperonimia existentes en la literatura. Además, se extrajeron los patrones presentes en el corpus de entrenamiento proporcionado por *SemEval* que contiene un extracto de texto donde las relaciones entre frases clave deben ser identificadas, y un archivo de anotaciones con frases clave identificadas y relaciones semánticas entre cada una frase. Posteriormente se convirtieron a expresiones regulares y se generó una lista de

palabras que cumplen con el patrón sin repetición y se procedió con la búsqueda de las relaciones para cada frase clave previamente extraída.

3.1 Extracción de patrones

En la literatura se encuentran los denominados patrones léxico sintácticos. Los cuales tienen la capacidad de extraer las relaciones presentes en un texto. Sin embargo, dado que el corpus proporcionado por *SemEval* es de dominio científico se identifica la necesidad de contar con patrones adicionales para este propósito. Por lo que se extrajeron patrones léxico sintácticos presentes en el corpus de entrenamiento. Para ello se proponen los siguientes pasos:

1. Obtener relaciones confirmadas. Partiendo del conjunto de datos de entrenamiento, se obtienen un par de frases con una relación semántica confirmada de sinonimia o hiponimia
2. Obtener patrones. Del texto extraído en el paso 1, se extraen del corpus las frases clave presentes en este obteniendo un conjunto del tipo:

<*fraseClave*>conector<*fraseClave*> donde conector se extrae como patrón para extraer una relación.

Finalmente se obtuvieron un total de 22 conectores para relaciones de sinonimia e hiponimia que se toman como patrones como se muestra en la Tabla 1. Los patrones aquí expuestos y los existentes en la literatura se usan en este trabajo.

Tabla 1 Patrones extraídos

Patrón	**Patrón**	**Patrón**	**Patrón**	**Patrón**	**Patrón**	**Patrón**	**Patrón**
S_1 a S_2	S_1 is often referred to as a S_2	S_1 or S_2	S_1 combining quantum features with a S_2	S_1 of which the S_2	S_1 within the so-called S_2	S_1 is found to be S_2	S_1 is the most important S_2
S_1 i.e. excitation energies and S_2	S_1 i.e., the S_2	S_1 such as steam generator tubes S_2	S_1 for applications S_2	S_1 such as the S_2	S_1 i.e. ones that are S_2	S_1 based on the S_2	S_1 based on a S_2 S_2
S_1 in S_2	S_1 the stability and error estimate of the S_2	S_1 the extrapolation method of S_2	S_1 defined at the same points leading to a S_2	S_1 located 500km from the lower and left side has a S_2	*S_1 (S_2)*		

3.2 Extracción de frases clave

Se extrajeron las frases clave por medio de *n*-gramas (*n*=1,2,3,4 y 5) y un pesado *tf-idf* de términos que aparezcan en el 50% de los documentos. El resultado serán las frases clave de cada documento que forma al corpus. La librería *sklearn* proporciona la función para el pesado *tf-idf* y los *n*-gramas. Para ello se proponen los siguientes pasos:

1. Lectura de corpus. El corpus se codificó en *utf-8*. El corpus fue segmentado en oraciones.
2. Generar *n*-gramas. Se generan *n*-gramas (*n*=1,2,3,4 y 5).
3. Pesado de términos. Se aplica la medida *idf* para usar la frecuencia de documento inversa. Se genera un diccionario de *frasesClave*: pesado *tf-idf*.
4. Clasificación de frases por sus pesos. Los *n*-gramas se ordenan en el diccionario según el peso *tf-idf* en orden descendiente.
5. Etiquetado: Las frases clave obtenidas fueron llevadas al etiquetador *Freeling* (Padró et al., 2012) para eliminar palabras unitarias con las etiquetas "IN" o "CC"

Las frases clave resultantes por cada documento fueron almacenadas para buscar su posición final e inicial en el corpus y de esta manera evaluar el desempeño del modelo propuesto.

3.3 Extracción de relaciones semánticas

Previamente se construyó un repositorio de patrones léxico sintácticos existentes en la literatura para extraer patrones de un corpus de Wikipedia. Los pasos que se siguieron para la extracción de estas relaciones semánticas en el corpus proporcionado por *SemEval* 2017 son:

1. Uso de un repositorio de patrones léxico sintácticos para las relaciones semánticas previamente creado. Cada patrón léxico sintáctico se convierte a expresiones regulares incluidos los extraídos en la sección 3.1.
2. El corpus se preprocesa eliminando elementos no *ascii* y se convierten mayúsculas a minúsculas.
3. Aplicar las expresiones regulares previamente obtenidas en el corpus pre-procesado generando una lista de palabras que cumplen con el patrón sin repetición.
4. Con las frases clave previamente identificadas en el paso 3.2 y las relaciones semánticas identificadas se lleva a cabo un mapeo y se extraen el tipo de relación según corresponda.

4 Conjunto de datos

Los datos proporcionados por *SemEval* 2017 tarea 10, son 500 artículos científicos del área de Ciencias de la Computación, Ciencias de Materiales y Física. El total de frases clave del gold estándar es de 2051. Cada uno de los 500 artículos fue extraído de la página de ScienceDirect y consta de tres archivos, el primero con todo el texto (.xml), el segundo con un párrafo del texto (.txt) y el tercero de anotaciones (.ann), el archivo contiene las frases clave con un identificador, su clasificación, la posición dentro del texto y la frase clave. Todo el conjunto de datos se encuentra en el idioma inglés. Las relaciones de tipo hiponimia incluidas en los datos proporcionados fue 123 para desarrollo, 418 para entrenamiento y 95 para pruebas. Para relaciones de tipo sinonimia 45 para desarrollo, 253 para entrenamiento y 112 para pruebas.

5 Resultados experimentales

SemEval proporciona un script para llevar a cabo la evaluación de las tareas desarrolladas. Las medidas de evaluación son precisión, exhaustividad y medida-F_1 que darán como resultado el rendimiento de la extracción de las frases clave y de relaciones semánticas.
La Tabla 2 expone algunos de los resultados obtenidos al aplicar los pasos descritos en la sección 3.2. Las frases clave expuestas son de longitud 1, 2, 3, 4 y 5.

Tabla 2 Ejemplos de frases clave extraídas

	Frase Clave	**Frase Clave**	**Frase Clave**
Dimensión 1	Cu40Z	hydrozicite	hydrozicite
Dimensión 2	Python tool	Data analysis	Electron microscopy
Dimensión 3	Diamond polished surface	The corrosion behavior	Preferential cathodic site
Dimensión 4	Kernel extensional matrix matching	Load path which contain	Probabilistic roulette wheel music
Dimensión 5	Fetch results of completed jobs	Efficiency of agricultural silage production	Organize and curate simulation data

La Tabla 3 muestra dos ejemplos de frases clave con el tipo de relación encontrada obtenidas al aplicar los pasos descritos en la sección 3.3.

Tabla 3 Ejemplos de relaciones encontradas

Frase Clave	**Frase Clave**	**Relación**
Cu40Z	Hydrozicite	Sinónimos
Arabinoxylan	Non starchy polysaccharides	Hipónimos

5.1 Evaluación

El método para la extracción de frases clave (subtarea 1) proporcionó un total de 173,162 elementos. Los resultados obtenidos se muestran en la Tabla 4. Como se observa los resultados son bajos en comparación con los obtenidos por otros autores en la literatura con métodos empleados en las mismas tareas (como se muestran en la Tabla 5). En la Tabla 5 se muestran los resultados obtenidos con la métrica-F_1 de 4 autores diferentes de 17 autores que participaron en *SemEval* 2017.

Tabla 4 Resultados para la extracción de frases clave

	Precisión	**Exactitud**	**Medida-F_1**
Frases Clave	0.02	0.34	0.04

Tabla 5 Resultados para la extracción de frases clave por participantes *SemEval* 2017

Autores	**Medida-F_1**
(Ammar et al., 2017)	0.55
(Kim et al., 2010)	0.56
(Tsujimura et al., 2017)	0.5
(Wang et al., 2017)	0.51

Para la extracción de relaciones semánticas (subtarea 3) los resultados son menores en comparación con los obtenidos en la literatura como se observa en la Tabla 6 donde se exponen los resultados obtenidos por otros autores en la misma tarea. En ambos casos la evaluación se realizó con el script y el *gold-standard* proporcionado por *SemEval*. La Tabla 7 muestra los resultados obtenidos al evaluar el método para la extracción de relaciones propuesto.

Tabla 6 Resultados para la extracción de relaciones semánticas por participantes *SemEval* 2017

Autores	**Medida-F_1**
(Ammar et al., 2017)	0.28
(Kim et al., 2010)	0.21
(Tsujimura et al., 2017)	0.19
(Wang et al., 2017)	0.2

Tabla 7 Resultados de la extracción de relaciones semánticas

	Precisión	**Exactitud**	**Medida-F_1**
Sinónimos	0.20	0.15	0.16
Hipónimos	0.24	0.13	0.17

6 Conclusiones

Este trabajo expone el método propuesto para: extracción de frases clave y relaciones semánticas entre ellas. Previamente se generó un repositorio de patrones léxico sintácticos presentes en la literatura, por ejemplo, los patrones de Hearst (Hearst, 1992) para relaciones de tipo hipónimo-hiperónimo. Primero se extrajeron las frases clave posteriormente se hizo un mapeo sobre el corpus y se extraen los conectores entre cada frase clave. Los conectores se consideraron como patrones y se agregaron al repositorio previamente generado. Los patrones obtenidos son transformados en expresiones regulares para obtener las relaciones existentes en el corpus. El propósito es identificar las frases clave para llevar a cabo un mapeo entre ellas y las relaciones semánticas entre ellas.

Las principales contribuciones de este trabajo son: un enfoque para obtener de manera automática patrones para descubrir relaciones de tipo sinonimia e hiponimia en documentos de dominio científico; y el enriquecimiento de un repositorio de patrones léxico sintácticos con el fin de ampliar el dominio que los patrones deben de cubrir.

El proceso de evaluación para la extracción de frases clave arrojó resultados bajos en comparación con los obtenidos en la literatura. Este hecho llevo a hacer una revisión manual en el 25% de los documentos y se observó la existencia de frases clave pero se llegó a la conclusión de que es necesario aplicar: un preprocesamiento adicional con el fin de eliminar palabras cerradas que generan información errónea, aplicar un modelo de aprendizaje profundo con el objetivo de mejorar los niveles de precisión y contar con un diccionario pre-existente de términos matemáticos con el fin de preservar esa información que en este trabajo fue cambiada lo que fue una de las causas que influyó en los resultados obtenidos. Por otro lado, se observó también que el etiquetador *Freeling* generó cambios en los verbos como "*be*" por "*been*" lo que provocó la presencia de frases clave con verbos conjugados y por lo tanto generó una diferencia con los archivos de evaluación.
El proceso de extracción de relaciones proporcionó resultados bajos, sin embargo, dado que las relaciones son en su mayoría extraídas con los patrones de la literatura y no están guiados a un dominio en específico se considera que extraer más patrones de otros corpus del mismo dominio enriquecerá y aumentará la lista de 22 patrones que se obtuvo en este trabajo y por lo tanto mejorará los resultados obtenidos. Además, se considera que contar con un diccionario de términos en este dominio también proporcionará resultados mayores a los obtenidos hasta ahora.

Agradecimientos

Los autores agradecen al Laboratorio Nacional de Supercómputo del Sureste de México (LNS), perteneciente al padrón de laboratorios nacionales CONACYT, por los recursos computacionales, el apoyo y la asistencia técnica brindados, a través del proyecto No 202103090C.

Referencias

Platero, J. M. G. (2019). Polisemia y monosémia en el léxico. Homonimia, sinonimia y antonimia. Liceus, Servicios de Gestión, pp. 2-9

Aggarwal, C. C. (2018). *Neural networks and deep learning*. Springer, 10, 978-3.

Akhtyamova, L., Alexandrov, M., & Cardiff, J. (2017). *Adverse drug extraction in twitter data using convolutional neural network*. In 2017 28th International Workshop on Database and Expert Systems Applications (DEXA) (pp. 88-92). IEEE.

Al-Zaidy, R. A., & Giles, C. L. (2018). *Extracting semantic relations for scholarly knowledge base construction*. In 2018 IEEE 12th international conference on semantic computing (ICSC) (pp. 56-63). IEEE.

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). *Understanding of a convolutional neural network*. In 2017 international conference on engineering and technology (ICET) (pp. 1-6). IEEE.

Bentrcia, R., Zidat, S., & Marir, F. (2018). *Extracting semantic relations from the Quranic Arabic based on Arabic conjunctive patterns*. Journal of King Saud University-Computer and Information Sciences, 30(3), pp. 382-390.

León-Araúz, P., San Martín, A., & Faber, P. (2016). *Pattern-based word sketches for the extraction of semantic relations*. In Proceedings of the 5th international workshop on computational terminology (Computerm2016) pp. 73-82.

Shanidze, O., & Petrasova, S. V. (2019). *Extraction of Semantic Relations from Wikipedia Text Corpus* (Doctoral dissertation).

Ta, C. D., & Thi, T. P. (2016). *Automatic extraction of semantic relations from text documents*. In International Conference on Future Data and Security Engineering, pp. 344-351. Springer, Cham.

Zhang, L., Hu, J., Xu, Q., Li, F., Rao, G., & Tao, C. (2020). *A semantic relationship mining method among disorders, genes, and drugs from different biomedical datasets*. BMC Medical Informatics and Decision Making, 20(4), pp. 1-11.

Vuotto, A., Bogetti, C., & Fernández, G. (2015). Aplicación del factor *TF-IDF* en el análisis semántico de una colección documental. Biblios, (60), pp. 1-13.

Sidorov, G. (2013). *N*-gramas sintácticos no-continuos. Polibits, 48, 69-78.

Ammar, W., Peters, M. E., Bhagavatula, C., & Power, R. (2017). *The ai2 system at semeval-2017 task 10 (scienceie): semi-supervised end-to-end entity and relation extraction*. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 592-596.

Kim, S. N., Baldwin, T., & Kan, M. Y. (2010). *Evaluating n-gram based evaluation metrics for automatic keyphrase extraction*. In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010) pp. 572-580.

Tsujimura, T., Miwa, M., & Sasaki, Y. (2017). *Tti-coin at semeval-2017 task 10: Investigating embeddings for end-to-end relation extraction from scientific papers*. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 985-989.

Wang, L., & Li, S. (2017). *PKU_ICL at SemEval-2017 task 10: Keyphrase extraction with model ensemble and external knowledge*. In Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017), pp. 934-937.

Padró, L., & Stanilovsky, E. (2012). *Freeling 3.0: Towards wider multilinguality*. In LREC2012.

Augenstein, I., Riedel, S., Vikraman, L., McCallum, A., & Das, M. (2017). SemEval-2017 task 10: Extracting keyphrases and relations from scientific publications. In The 11th International Workshop on Semantic Evaluation (SemEval-2017). Vancouver, Canada: Association for Computational Linguistics.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics.

Capítulo 2

Método para la Evaluación de Medidas de Similitud Semántica utilizando Textos Cortos

Maricela Bravo[1], Luis Fernando Hoyos Reyes[1], Domingo Rodríguez Benavides[1]

[1] Universidad Autónoma Metropolitana
Departamento de Sistemas

mcbc@azc.uam.mx, hrlf@azc.uam.mx, dorobe@azc.uam.mx

Resumen. Existen múltiples colecciones de artículos científicos en línea y bases de datos disponibles públicamente, para aprovechar al máximo estos recursos es necesario procesar, organizar y correlacionar los textos con respecto a una clasificación o taxonomía. Existen diversas medidas de similitud semántica que se aplican entre textos cortos, las cuales permiten lograr una organización eficiente y una correlación más relevante entre los textos. Sin embargo, determinar el mejor método para calcular la similitud entre textos es una tarea ardua, ya que hay muchas medidas de similitud reportadas en la literatura. Además, debe considerarse la recopilación de textos a los que se aplican las medidas de similitud; si bien algunas medidas son útiles para algunos tipos de fuentes de información, fallan cuando cambia la colección de datos. Por lo tanto, es necesario contar con un método para evaluar el desempeño de las medidas de similitud desde una perspectiva estadística y en términos de la precisión alcanzada por cada medida.

Palabras clave: Medidas de similitud semántica, comparación de textos cortos, clasificación de la publicación científica, evaluación de medidas de similitud.

1 Introducción

Existe un creciente interés por la correlación semántica (o indexación) de las publicaciones científicas con respecto a taxonomía de "temas o tópicos de investigación" como por ejemplo la clasificación ACM. Para establecer una correlación existe la necesidad de comparar y calcular similitudes entre publicaciones y conceptos. El objetivo de las medidas de similitud es descubrir la relación o distancia semántica que existe entre una publicación con los temas de una taxonomía, empleando los textos extraídos del título o del resumen. Existen muchas medidas de similitud semántica aplicables a textos cortos; entre estas, las que se utilizan en este artículo son las medidas semánticas basadas en WordNet (Wu y Palmer, 1994), (Jiang y Conrath, 1997), (Leacock y Chodorow, 1998), (Lin, 1998) y (Resnik, 1995).

Sin embargo, la tarea de calcular similitudes semánticas entre títulos y temas implica un problema de dimensionalidad y escalabilidad. Según una estimación presentada por (Jinha, 2010) hubo 50 millones de publicaciones y aproximadamente 2,5 millones se producen por año.

1.1 Formulación del Problema

Dado un conjunto de publicaciones ***P***, una taxonomía de conceptos ***T***, un conjunto de medidas de similitud ***S*** y un conjunto de criterios de evaluación ***E***, el cálculo de similitudes se realiza con base a $<P, T, S, E>$
Dónde
P representa una colección de ***i*** publicaciones $P = \{p_1, p_2, p_3, ..., p_i\}, i > 0$
T representa una taxonomía de temas que organiza ***j*** temas $T = \{t_1, t_2, t_3, ..., t_j\}, j > 0$
S representa un conjunto de ***k*** medidas de similitud $S = \{s_1, s_2, ..., s_k\}\ k > 0$
E representa los criterios de evaluación $E = \{e_1, e_2, ..., e_l\}, l > 0$

El número de cálculos de similitud se define por ***calc*** = $i * j * k$, y el tiempo requerido para este número de cálculos es necesario incorporar el tiempo de ejecución de cada medida de similitud, ***tiempo*** = $i * j *$ ($k *$ *tiempo* ($k *$ tiempo (k)). Donde *el tiempo* (k) depende del algoritmo particular y los recursos computacionales requeridos por los cálculos de similitud. Suponiendo que hay una colección de 2000 publicaciones, 2000 temas y seis medidas de similitud, el número de cálculos es ***calc*** = 2000 * 2000 * 6, es decir, 24 millones de cálculos multiplicados por el tiempo requerido para cada una de las seis medidas de similitud. Teniendo en cuenta que se puede utilizar más de un criterio de evaluación para la selección de medidas de similitud, el problema de investigación se divide en dos objetivos.

a) Desarrollar un método para reducir el número de cálculos de similitud, el cual permita comparar la eficiencia de cada medida y decidir sobre la medida que mejores resultados arroje para la colección de publicaciones de entrada.
b) Definir una función de evaluación que incorpore los criterios para seleccionar la mejor medida de similitud.

En este trabajo, se presenta un método sistemático que permite analizar un conjunto de medidas de similitud con ejemplares de la colección de publicaciones, de tal forma que se puede conocer cual medida de similitud es la que devuelve mejores resultados con respecto a criterios de evaluación. Este método permite obtener información sobre cuál medida será suficiente para determinar la distancia semántica para un gran volumen de publicaciones, sin la necesidad de que se realicen todos los cálculos con todas las medidas de similitud. El objetivo de este método es proporcionar evidencia para la toma de decisiones. Este método muy bien puede aplicarse en la toma de decisiones para otras áreas de la Inteligencia Artificial, por ejemplo, para determinar la función heurística que de mejores resultados en algoritmos de aprendizaje.

El resto del artículo se encuentra organizado de la siguiente forma: en la sección 2 se describe el método, en la sección 3 se realiza una descripción de las fuentes de información y la taxonomía empleados, en la sección 4 se describen las medidas de similitud semánticas empleadas en los cálculos, en la sección 5 se presenta la experimentación y evaluación de resultados, finalmente en la sección 6 se presentan las conclusiones y trabajos futuros.

2 Descripción del Método

Con el objetivo de reducir el número de cálculos y el tiempo necesario para la evaluación de las medidas de similitud, el método propuesto (que se muestra en la Figura 1) consta de los siguientes pasos:

1) ***Recopilar datos*** de entrada: seleccionar una colección de publicaciones de investigación con las que trabajar y seleccionar una taxonomía de temas de investigación.
2) ***Preprocesar los textos***: es importante verificar que los textos de los títulos de las publicaciones no estén vacíos, que no contengan caracteres mal formateados, y crear una bolsa de palabras que representen los textos.
3) ***Seleccionar las medidas*** de similitud: el conjunto de cálculos de similitud son métodos bien documentados para la evaluación de la similitud de textos cortos. Sin embargo, durante la última década se han reportado numerosos métodos de comparación entre textos, entre estos se encuentran los basados en el uso del diccionario WordNet. Es importante seleccionar las medidas que son de interés para el cálculo, ya que estos cálculos requieren recursos computacionales.
4) ***Experimentación***: determinar el tamaño de la muestra y seleccionar aleatoriamente muestras de títulos de publicaciones y temas de la taxonomía para ejecutar los cálculos. Repita el experimento ***n*** veces.
5) ***Evaluación***: determinar los criterios de evaluación que se utilizarán, y formular una función de evaluación que incorpore los criterios y posibles ponderaciones. Aplicar la función de evaluación y concentrar los resultados para determinar la medida de similitud que mejor se ajuste a los criterios de evaluación.

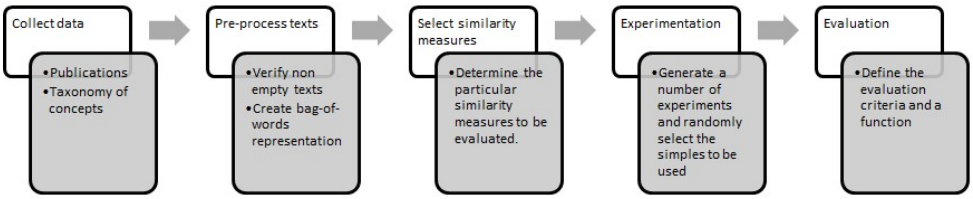


Figura 1. Método propuesto para la evaluación de medidas de similitud.

3 Recopilación de datos

3.1 Bases de Datos Bibliográficas

Con respecto a los artículos académicos y las publicaciones de investigación, hay muchas bases de datos bibliográficas disponibles en línea, por ejemplo: ArnetMiner[1], CiteULike[2], CiteSeer[3], la Colección de Bibliografías de Ciencias de la Computación[4], la base de datos de publicaciones de Ciencias de la Computación DBLP[5], la Biblioteca Digital de la ACM[6], IEEE Xplore[7] y Microsoft Academic[8]. La colección de datos bibliográficos de ArnetMiner fue seleccionada porque ofrece una colección de publicaciones en un formato de datos estructurados con información sobre autores, publicaciones y citas. En particular, para este trabajo, se utilizó como datos de entrada el archivo AMiner-Paper.rar incluido en la colección de la Red Social Académica que contiene 2.092.356 artículos (ver Figura 2), que incluye registros con una identificación de cada artículo, el título del artículo, la lista de autores, las afiliaciones de los autores, el año de publicación, el lugar de publicación, la lista de referencias y el resumen. El proyecto ArnetMiner está diseñado para buscar y realizar minería de datos de publicaciones académicas en Internet, utilizando el análisis de redes sociales para identificar colaboraciones entre investigadores, conferencias y publicaciones. ArnetMiner tiene como objetivo proporcionar servicios como búsqueda de expertos, búsqueda por regiones, recomendación de revisores, búsqueda de asociaciones, búsqueda de cursos, evaluación del rendimiento académico y modelado de temas.

```
#index ---- identificación de este documento
#* ---- título del artículo
#@ ---- autores (separados por punto y coma)
#o ---- afiliaciones (separadas por punto y coma)
#t ---- año
#c ---- ha llegado la publicación
#% ---- el identificador de referencias de este trabajo
#! resumen de ----
```

Figura 2. Estructura del archivo AMiner

[1] https://aminer.org/
[2] https://citeulike.org/
[3] https://citeseerx.ist.psu.edu/
[4] https://liinwww.ira.uka.de/bibliography/
[5] https://dblp.uni-trier.de/
[6] https://dl.acm.org/
[7] https://ieeexplore.ieee.org/Xplore/
[8] https://academic.microsoft.com/

3.2 Taxonomía de Tópicos de Investigación

De acuerdo con (Lambe, 2014) una taxonomía se puede definir como un conjunto estructurado de nombres y descripciones utilizados para organizar la información y los documentos de una manera consistente. Las taxonomías son cruciales para la gestión de las organizaciones. Según (Pincher, 2010) todos los tipos de sistemas de gestión en una organización son casi inútiles si no utilizan taxonomías. Las taxonomías son necesarias para organizar el almacenamiento y la administración de los recursos, y para apoyar una mejor búsqueda de recursos. Las siguientes son algunos ejemplos de taxonomías de conocimiento:

a) La ACM Computing Classification System, es un sistema de clasificación estándar para el campo del conocimiento de las Ciencias de la Computación. Es mantenido por la Organización ACM.[9]
b) La Ontología de Ciencias de la Computación (CSO), es una ontología a gran escala, generada automáticamente de áreas de investigación en el campo de la Informática, que incluye alrededor de 15,000 temas y 70,000 relaciones semánticas. [10]
c) En las áreas de investigación relacionadas con la Física y la Astronomía, la taxonomía más popular utilizada es el Esquema de Clasificación de Física y Astronomía (PACS). PACS fue desarrollado en 1970 por el Instituto Americano de Física (AIP) para clasificar la literatura científica utilizando un conjunto jerárquico de códigos.
d) La Clasificación de Asignaturas de Matemáticas (MSC) es la principal taxonomía utilizada en el campo de las Matemáticas. Esta taxonomía es mantenida por Mathematical Reviews (MRDB) y Zentralblatt MATH (ZMATH).[11]
e) El Medical Subject Heading (MeSH) es un vocabulario controlado producido por la Biblioteca Nacional de Medicina, se utiliza para indexar, catalogar y buscar conceptos y documentos biomédicos y relacionados con la salud.[12]

Para facilitar la tarea de búsqueda de los autores que abordan un tema en particular, o para recuperar un conjunto de publicaciones que están estrechamente relacionadas con un tema de investigación en particular, es necesario correlacionar las publicaciones con una taxonomía del conocimiento, relacionada con el área de conocimiento de interés. En este artículo, el conjunto de publicaciones estará correlacionado y organizado con respecto al área de conocimiento de Ciencias de la Computación, por lo que se seleccionó el Sistema de Clasificación de Computación ACM.

[9] https://www.acm.org/publications/class-2012
[10] http://skm.kmi.open.ac.uk/cso/
[11] https://mathscinet.ams.org/msc/msc2010.html
[12] https://meshb.nlm.nih.gov/search

4 Medidas de Similitud Semántica

En los últimos años ha habido un creciente interés en la investigación y desarrollo de métodos para calcular la similitud semántica de textos cortos. Por ejemplo, el International Workshop of Semantic Evaluation (*SemEval*[13]) ha abordado la tarea de la Similitud Textual Semántica (STS) en 2012, 2014, 2015 y 2017. En la literatura especializada se han reportado diversos enfoques: métodos sintácticos, semánticos, pragmáticos, probabilísticos, entre otros. De particular interés son las medidas basadas en conocimiento que utilizan la base de datos léxica wordnet. En el método que se presenta en este artículo se utilizaron las siguientes medidas para calcular la similitud semántica entre los títulos y las clasificaciones ACM. Estas medidas de similitud semántica utilizan la base de datos de WordNet y explotan relaciones adicionales no jerárquicas.

(Wu y Palmer, 1994) introdujeron una medida de similitud que encuentra la longitud del camino hacia el nodo raíz desde la subclase menos común (LCS) de dos conceptos, que es el concepto común más específico que comparten como antepasado. Este valor se escala mediante la suma de las longitudes de trazado desde los conceptos individuales hasta la raíz.

(Jiang y Conrath, 1997) describen una medida de similitud semántica basada que utiliza la probabilidad condicional de encontrar una instancia de un conjunto de sinónimos de subclase dada una instancia de un conjunto de sinónimos de superclase. Así se considera el contenido de información de los dos nodos, así como el de su subclase más específica.

La relación semántica de (Leacock y Chodorow, 1998) es una medida que encuentra la longitud de ruta más corta entre dos conceptos, y escala ese valor por la longitud máxima de la ruta en la jerarquía is-A en la que ocurren. Considera que la distancia conceptual entre dos nodos es proporcional al número de aristas que separan los dos nodos de la jerarquía.

La relación semántica de (Leacock y Chodorow, 1998) es un esquema de conteo de nodos conocido como PATH. La puntuación de similitud es inversamente proporcional al número de nodos a lo largo de la ruta más corta entre los conjuntos de sinónimos. La ruta más corta posible ocurre cuando los dos conjuntos de sinónimos son iguales, en cuyo caso la longitud es 1. Por lo tanto, el valor máximo de relación es 1.

(Lin, 1998) presenta una medida que calcula la relación semántica entre dos conceptos. Lin declaró que "la similitud entre A y B se mide por la relación entre la cantidad de información necesaria para establecer la similitud de A y B y la información necesaria para describir completamente lo que son A y B". Esta medida utiliza la cantidad de información necesaria para establecer la similitud entre los dos conceptos y la información necesaria para describir estos términos.

(Resnik, 1995) presenta un enfoque de similitud semántica que utiliza la información de los conceptos, calculada a partir de su frecuencia de ocurrencia en un gran corpus. Considera que la similitud entre un par de conceptos puede juzgarse por "la medida en que

[13] https://semeval.github.io/SemEval2022/

comparten información", Resnik calcula la relación semántica entre dos conceptos lexicalizados.

Existen otras aplicaciones que utilizan las relaciones del diccionario de WordNet, por ejemplo en (Hirst, y St-Onge, 1998) se describen las cadenas léxicas, las cuales proporcionan suficiente contexto para resolver las ambigüedades léxicas. Asimismo, en (Boom, et al., 2015), se presenta un análisis de mediciones de similitud aplicables en textos cortos.

5 Experimentación y Evaluación

Para la experimentación se utilizaron dos fuentes de datos: una colección de publicaciones de 2018 extraídas del archivo AMiner-Paper.rar, una colección de 2500 temas de investigación de la clasificación ACM y un conjunto de 6 medidas de similitud. Teniendo en cuenta el tamaño de estas colecciones, el número de cálculos de similitud es el producto de 2081 títulos, 2500 temas de investigación y 6 medidas de similitud, es decir, un total de 31,215,000 cálculos de similitud. El primer objetivo es reducir el número de cálculos de similitud y el tiempo requerido para su ejecución. La Figura 3 muestra el proceso de generación de 10 archivos de muestra con cálculos de similitud. El proceso comienza seleccionando aleatoriamente una muestra de 100 temas de la taxonomía de ACM y 100 títulos de publicación del conjunto de datos de publicación. Luego se calculan las seis medidas de similitud entre todos los pares de temas y títulos. Para filtrar los resultados representativos, la media de todas las medidas se utiliza para seleccionar aquellas similitudes que son superiores a 0.4. Para la generación de los 10 archivos de ejemplares se ejecutó 10 veces el proceso que se muestra en la Figura 3.

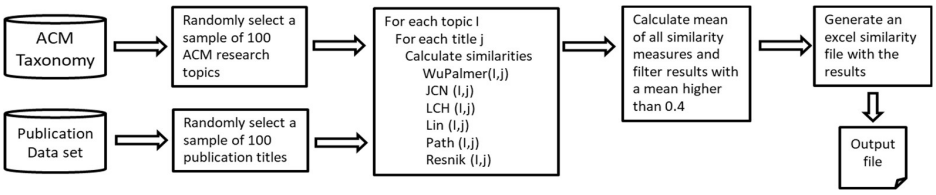


Figura 3. Proceso para generar aleatoriamente archivos de muestra de similitud.

Para determinar cuál es la medida de similitud que genera los mejores resultados, se establecieron dos criterios de evaluación: un análisis estadístico de las medidas y el desempeño de las medidas. Estas evaluaciones se describen en las siguientes subsecciones.

5.1 Análisis Estadístico Exploratorio de Medidas de Similitud

El objetivo de este análisis es determinar la estabilidad de las medidas de similitud bajo un criterio de varianza. La Tabla 1 muestra los resultados del análisis estadístico de las medidas para los 10 archivos de muestra generados. Estos archivos fueron generados ejecutando 10 veces el proceso que se describe en la Figura 3.
StatEval es una medida que permite calcular el error en base a la varianza estadística de los datos. El propósito de este cálculo es seleccionar la medida de similitud que devuelve el menor error.

Tabla 1. Análisis estadístico exploratorio de medidas de similitud.

Archivos de similitudes	WuPalmer	JCN	LCH	Lin	Path	Resnik
Ejemplo 1	0.51619336	0.17166925	0.86420904	0.25045354	0.21087887	0.65576238
Ejemplo 2	0.52538175	0.17523732	0.82150527	0.26374271	0.2214339	0.72224
Ejemplo 3	0.51461753	0.2018436	0.84657772	0.2750127	0.24099431	0.65534076
Ejemplo 4	0.4525342	0.26407365	0.70802012	0.3126669	0.27430433	0.63055835
Ejemplo 5	0.47479922	0.25382032	0.73659554	0.34159844	0.27176552	0.6755739
Ejemplo 6	0.50657947	0.13513243	0.92798221	0.23316118	0.17964823	0.76025746
Ejemplo 7	0.52282548	0.15522012	0.85554766	0.26629569	0.19318727	0.78180292
Ejemplo 8	0.51194417	0.19981332	0.80169707	0.27086563	0.24399806	0.64674605
Ejemplo 9	0.49068779	0.26529114	0.75884988	0.30640834	0.29640042	0.60478231
Ejemplo 10	0.50109492	0.15704171	0.82687845	0.24158274	0.19112075	0.71912867
Significar	0.50166579	0.19791429	0.8147863	0.27617879	0.23237317	0.68521928
Varianza	0.00053059	0.00229991	0.00432583	0.00116845	0.00157976	0.00336345
Desviación estándar	0.0230346	0.04795739	0.06577106	0.0341826	0.03974617	0.05799529
Intervalo	0.01603248	0.03337917	0.0457778	0.02379168	0.02766402	0.04036573
StatEval	**2.661513197**	14.04561751	**4.679005884**	7.174261546	9.914520739	4.905978605

De acuerdo con los resultados de la Tabla 1, la medida de similitud más estable es la similitud de WuPalmer, seguida de la medida LCH porque obtienen los errores relativos más pequeños. Cabe mencionar que una buena cota superior de el promedio de la muestra es el límite superior del intervalo de confianza para un nivel de certeza del 90%. Por ejemplo para WuPalmer fue de 0.501665789 y la cota máxima fue de 0.51501769 con un nivel de certeza del 90%.

5.2 Rendimiento de las Medidas

La evaluación del rendimiento de las medidas de similitud se realizará utilizando las medidas *Precision*, *Recall* y *F1*. La Tabla 2 muestra el cálculo de la *Precisión* de cada medida de similitud semántica, aplicada a cada archivo de muestra. En consecuencia, las medidas Lin y Path muestran mejores resultados de *Precisión* que las demás. La Tabla 3 muestra los resultados de *Cobertura* de cada medida.

Tabla 2. Resultados del cálculo de la *Precisión* de similitudes para cada muestra.

Archivos de similitudes	Número de comparaciones	WuPalmer	JCN	LCH	Lin	Path	Resnik
Ejemplo 1	23	0.0909	0.5000	0.0870	1.0000	1.0000	0.1176
Ejemplo 2	30	0.2857	0.0000	0.2759	0.0000	0.0000	0.2692
Ejemplo 3	60	0.0690	0.0000	0.0667	0.0000	0.0000	0.0741
Ejemplo 4	12	0.0000	0.0000	0.0909	0.0000	0.0000	0.0909
Ejemplo 5	19	0.2000	1.0000	0.0526	1.0000	1.0000	0.0556
Ejemplo 6	55	0.0800	0.0000	0.0556	1.0000	1.0000	0.0408
Ejemplo 7	24	0.1667	0.0000	0.0833	0.0000	0.0000	0.0833
Ejemplo 8	20	0.1000	0.0000	0.0500	0.0000	0.0000	0.0526
Ejemplo 9	37	0.0769	0.0000	0.0541	0.0000	0.0000	0.0690
Ejemplo 10	17	0.0000	0.0000	0.0588	0.0000	0.0000	0.0625
Promedio	29.7	0.1069	0.1500	0.0875	**0.3000**	**0.3000**	0.0916

Tabla 3. Resultado del cálculo de la *Cobertura* de similitudes para cada muestra.

Archivos de similitudes	Número de comparaciones	WuPalmer	JCN	LCH	Lin	Path	Resnik
Ejemplo 1	23	0.5000	0.5000	1.0000	0.5000	0.5000	1.0000
Ejemplo 2	30	0.6667	0.0000	0.8889	0.0000	0.0000	0.7778
Ejemplo 3	60	0.5000	0.0000	1.0000	0.0000	0.0000	1.0000
Ejemplo 4	12	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000
Ejemplo 5	19	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Ejemplo 6	55	0.6667	0.0000	1.0000	0.3333	0.3333	0.6667
Ejemplo 7	24	1.0000	0.0000	1.0000	0.0000	0.0000	1.0000
Ejemplo 8	20	1.0000	0.0000	1.0000	0.0000	0.0000	1.0000
Ejemplo 9	37	0.5000	0.0000	1.0000	0.0000	0.0000	1.0000

Ejemplo 10	17	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000
Promedio	29.7	0.5833	0.1500	**0.9889**	0.1833	0.1833	**0.9445**

Como los resultados de los cálculos de similitud mostraron que hay un gran número de diferencias entre los conceptos y títulos de ACM, entonces es necesaria una medida que equilibre la *Precisión* y la *Cobertura*. La medida *F1* es la media armónica de las puntuaciones de *Precisión* y la *Cobertura*. La medida de *F1* penaliza a los clasificadores con puntajes de *Precisión* y *Cobertura* desbalanceados. La Tabla 4 y la Figura 4 muestran que las medidas con los mejores resultados *F1* son las de Lin y Path.

Tabla 4. Resultados del cálculo de la medida *F1*.

	WuPalmer	JCN	LCH	Lin	Path	Resnik
Precisión	0.1069	0.1500	0.0875	0.3000	0.3000	0.0916
Cobertura	0.5833	0.1500	0.9889	0.1833	0.1833	0.9445
Medida F1	0.1807	0.1500	0.1608	**0.2276**	**0.2276**	0.1669

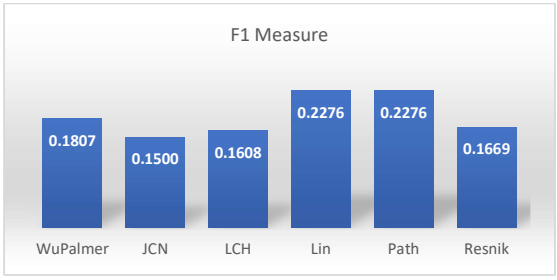


Figura 4. Puntuaciones de *Precisión* y *Cobertura* de las medidas de similitud.

5.3 Evaluación Global de las Medidas

La evaluación global que se presenta en la Fórmula 1 se calcula como una media ponderada de la medida *F1* y el resultado del análisis exploratorio. Las ponderaciones w_1, w_2 se establecen de acuerdo al criterio del evaluador, éstas deben tomar valores entre 0 y 1, y la suma debe ser igual a 1. Para el cálculo de la media es importante observar que las dos mediciones tienen un significado inverso. Para la medida *F1*, cuanto mayor sea el valor devuelto, mejor es la medida; mientras que para el resultado del análisis exploratorio,

cuanto menor sea el valor, mejor. Por lo tanto, el cálculo del promedio global incluye el valor inverso del análisis exploratorio.

$$OverEval = (F1 * w_1) + \left(\frac{1}{StatEval} * w_2\right) \quad (1)$$

Dónde
F1 es la media armónica de las medidas de precisión y recuperación
StatEval es el error relativo de las medidas
w_1, w_2 representan los pesos con valores entre [0…1], tal que $w_1 + w_2 = 1$

Tabla 5. Precisión y recuperación de medidas de similitud.

	WuPalmer	JCN	LCH	Lin	Path	Resnik
StatEval	2.661513	14.045618	4.679006	7.174262	9.914521	4.905979
Medida F1	0.180700	0.150000	0.160800	0.227600	0.227600	0.166900
OverEval	**0.258710**	0.118479	0.181968	0.192315	0.176905	0.181673

La Tabla 5 muestra que la mejor medida de similitud es el WuPalmer, considerando el análisis estadístico exploratorio y la medida F1.

6 Conclusiones

En este artículo se describe un método para decidir qué medida de similitud es mejor con respecto al problema de calcular las distancias semánticas entre textos cortos. Este método es especialmente efectivo porque reduce el tamaño de los cálculos para grandes cantidades de publicaciones, y mediante una técnica de muestreo permite determinar la medida de similitud que ofrecerá los mejores resultados sobre el conjunto completo.

El uso de una medida de evaluación basada únicamente en la precisión y cobertura no permite determinar si una medida de similitud dará lugar a errores estadísticos con respecto a los datos particulares que se utilizan. En cambio, en este trabajo se emplea un método de evaluación combinado, que proporciona una referencia más adecuada para decidir sobre la medida de similitud más confiable para el conjunto de publicaciones específico.

Como trabajo a futuro se realizarán más experimentos de este método utilizando otros tipos de textos en inglés, se emplearán otras medidas de similitud, así como diferentes taxonomías.

Referencias

Jinha, A. E. (2010). Article 50 million: an estimate of the number of scholarly articles in existence. Learned Publishing, 23(3), 258-263. Recuperado de: https://onlinelibrary.wiley.com/doi/pdf/10.1087/20100308

Wu, Z., & Palmer, M. (1994, June). Verbs semantics and lexical selection. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics (pp. 133-138). Recuperado de: https://arxiv.org/pdf/cmp-lg/9406033.pdf

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings of International Conference on Research in Computational Linguistics, Taiwan. Recuperado de: https://arxiv.org/pdf/cmp-lg/9709008.pdf

Leacock, C., & Chodorow, M. (1998). Combining local context and WordNet similarity for word sense identification. WordNet: An electronic lexical database, 49(2), 265-283.

Lin, D. (1998, July). An information-theoretic definition of similarity. In Icml (Vol. 98, No. 1998, pp. 296-304). Recuperado de: http://dit.unitn.it/~p2p/RelatedWork/Matching/an-information-theoretic-definition.pdf

Philip Resnik. (1995). Using information content to evaluate semantic similarity. In Proceedings of the 14th International Joint Conference on Artificial Intelligence, pages 448–453, Montreal.

Lambe, P. (2014). Organising knowledge: taxonomies, knowledge and organisational effectiveness. Elsevier.

Pincher, M. (2010). A guide to developing taxonomies for effective data management. Computer Weekly, 8.

Hirst, G., & St-Onge, D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. WordNet: An electronic lexical database, 305, 305-332. Recuperado de: https://www.cs.swarthmore.edu/~richardw/classes/cs65/f08/litreview/meggie-malcolm.pdf

De Boom, C., Van Canneyt, S., Bohez, S., Demeester, T., & Dhoedt, B. (2015, November). Learning semantic similarity for very short texts. In 2015 ieee international conference on data mining workshop (icdmw) (pp. 1229-1234). IEEE. Recuperado de: https://ieeexplore.ieee.org/

Capítulo 3

Exploración de modelos pre-entrenados basados en BERT para el análisis de polaridad de tuits en español

Erick Barrios González, Mireya Tovar Vidal, Fernando Zacarias Flores, Pedro Bello López

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
Avenida San Claudio y 14 Sur, Ciudad Universitaria, Puebla 72570, México

erick.barrios@alumno.buap.mx, mireya.tovar@correo.buap.mx, fzflores@yahoo.com.mx, pedro.bello@correo.buap.mx

Resumen. En este artículo se revisa la implementación de dos modelos pre-entrenados basados en *BERT* ("*bert-base-multilingual-cased*" y "*IIC/beto-base-spanish-sqac*") para resolver la tarea 1.1 de "*Workshop on Semantic Analysis at SEPLN 2020*" (*TASS* 2020), esta tarea consiste en el análisis polaridad de tuits (*tweets*) en español de diferentes países hispanohablantes. También se evalúa los efectos del pre-procesamiento y sinónimos de palabras en las entradas para los modelos mencionados anteriormente. Esta investigación se realiza con la finalidad de encontrar los puntos a mejorar en el análisis polaridad de tuits (*tweets)*, principalmente en la manera en que los modelos pre-entrenados interpretan palabras que no se encuentran en su vocabulario, debido a variaciones en el lenguaje, expresiones regionales, faltas de ortografía, uso de emojis, etc.

Palabras Clave: Análisis de polaridad, *PLN*, *BERT*.

1 Introducción

En 2021 el español fue el segundo lenguaje con más hablantes nativos en el mundo y la segunda lengua más utilizada en redes sociales como *Facebook*, *Instagram*, *LinkedIn* y *Twitter* (Fernández, 2021).

El análisis de sentimientos o la extracción de opiniones o el análisis de polaridad son subáreas del procesamiento del lenguaje natural y están orientadas al estudio de los sentimientos, emociones y opiniones expresadas en un texto; estos textos se enfocan en un objeto (un producto, servicio, entidad, organización, tema o evento).

El análisis de polaridad tiene como objetivo detectar si un texto expresa una opinión positiva, negativa o neutral y se ha vuelto importante debido a que es una herramienta que nos dan más información acerca de la opinión de muchas personas.

TASS ("*Semantic Analysis at SEPLN*") apareció en 2012 y fue la primera tarea orientada al análisis de sentimientos en Twitter para el lenguaje español y desde 2019 hasta 2020 fue

parte de *IberLEF* (*Iberian Languages Evaluation Forum*). En 2020, el nombre de *TASS* cambió a "*Workshop on Semantic Analysis at SEPLN*", abarcando más tareas de procesamiento semántico.

Existen múltiples enfoques para resolver los problemas que plantea el análisis de sentimientos y el análisis de polaridad como muestra Valladares (2022), dentro de estos enfoques se encuentra el aprendizaje profundo, que es un tipo de aprendizaje automático. Actualmente el aprendizaje profundo es la principal estrategia para resolver el análisis de sentimientos (valladares, 2022).

El aprendizaje profundo consiste en utilizar redes neuronales (algoritmos inspirados en el funcionamiento del cerebro humano) para aprender tareas o aprender grandes cantidades de datos (Hernández, 2021).

La tarea 1.1 de *TASS* 2020 aborda el análisis de polaridad en tuits (*tweets*) en español y será la tarea a revisar en este artículo. El objetivo de esta tarea es la evaluación de sistemas de clasificación de polaridad de tuits escritos en español (incluyendo sus variantes).

La tarea se enfoca en clasificar individualmente cada variación del español según el país de donde sean los tuits. Para esta tarea se plantean las siguientes variantes por país: ES-España, PE-Perú, CR-Costa Rica, UR-Uruguay, MX-México. *TASS* proporciona un corpus de entrenamiento y uno de evaluación, además se permitió utilizar cualquier otro corpus o recurso lingüístico además de los proporcionados por *TASS* 2020.

Las etiquetas para la clasificación de los tuits son las siguientes:

- P: Positivo
- N: Negativo
- NEU: Neutral (Incluye tuits no clasificados)

En este artículo se exploran soluciones para la tarea 1.1 de *TASS* 2020, implementando modelos (*BERT*) diferentes a los que se mostrarán en el estado del arte, estos modelos han sido pre-entrenados con una mayor cantidad de información a los que se mostrarán en el estado del arte (contemplando un modelo especializado en múltiples lenguajes y un modelo especializado únicamente en el español). Se utilizará un enfoque orientado a la parte del pre-procesamiento de los tuits (explorando el reemplazo de palabras que no se encuentren en el vocabulario de cada modelo por sinónimos).

Este artículo se encuentra distribuido de la siguiente forma: Primero, en la sección 2 se revisa el estado del arte, en la sección 3 se muestra la solución propuesta y la metodología implementada, la sección 4 muestra los resultados experimentales y finalmente, en la sección 5 se muestran las conclusiones.

2 Estado del arte

En 2020 de los 14 participantes en *TASS* 2020 (García M. et al., 2020), los mejores resultados son de los participantes Palomino y Ochoa (2020), González J. et al (2020) y García J. et al. (2020).

Los autores Palomino y Ochoa (2020) proponen un sistema para mejorar la clasificación de polaridad en pequeños conjuntos de datos basados en un modelo de lenguaje de alto desempeño (LM) denominado *BERT* (*Bidirectional Encoder Representations from Transformers*) (variante de *BERT* para producir aumento de datos contextuales).

Mientras que García J. et al. (2020) (UMUTeam) implementaron tres ejecuciones. La primera ejecución (LF + WE) consistió en características lingüísticas entrenadas con un perceptrón multicapa en combinación con incrustaciones de palabras entrenadas con una red neuronal convolucional (*CNN*); la segunda ejecución (LF) utilizó las características lingüísticas entrenadas con *Support Vector Machines* (*SVM*), y la tercera ejecución (LF + SE) utilizó la combinación de características lingüísticas con incrustaciones de oraciones.

Finalmente, González J. et al (2020) (ELiRF-UPV) utilizan *Deep Averaging Networks* (*DAN*) como línea de base (estos modelos consisten en aplicar redes de avance sobre representaciones de texto basadas en incrustaciones de palabras promedio) y *TWilBERT* (un *framework* basado en *BERT* para entrenar, evaluar y ajustar modelos en el dominio de *Twitter*), el cual fue el mejor sistema implementado para el análisis de polaridad en *TASS* 2020. *TWilBERT* fue entrenado con 94 millones de pares de tuits (47M positivos y 47M negativos).

Revisando trabajos posteriores referentes a esta tarea encontramos que en el trabajo de Valladares (2022) hay una revisión del estado del arte de 2009 a 2021 para el análisis de sentimientos, mostrando que el aprendizaje automático es actualmente la herramienta más utilizada para estas tareas.

En el trabajo de López et al. (2021) se utiliza *BERT* para el análisis de sentimientos de comentarios de *Google Play*, comparándolo con el algoritmo *Support Vector Machine* (*SVM*) y el clasificador *Naïve Bayes*, mostrando que el modelo *BERT* obtiene mejores resultados.

En el trabajo de Mazo (2021) se compara el modelo *BERT* con los modelos *BILSTM* (*Bidirectional Long Short-Term Memory*), para la extracción de palabras clave para el análisis de sentimientos, los mejores resultados se obtuvieron con *BERT*.

También en el trabajo de Scola y Segura (2021) se compara *BERT* con un modelo basado en redes neuronales *LSTM* (*Long Short-Term Memory*), obteniendo mejores resultados con *BERT*.

Se puede observar en el trabajo de Pérez et al. (2021) que se comparan varios modelos basados en *BERT* para el análisis de sentimientos y emociones. Principalmente compara *BERTWEET* (Modelo basado en *BERT* especializado en el dominio de *Twitter*) y BETO (Modelo basado en *BERT* especializado en el idioma español).

Otra comparación entre modelos basados en *BERT* la encontramos en el trabajo de Zárate (2021), donde se realiza una comparación de varios modelos *BERT* pre-entrenados para español, obteniendo mejores resultados con "*BERT-base-spanish-wwm-uncased*".

Finalmente, en el trabajo de González I. (2020) podemos observar recomendaciones de características a cubrirse cuando se usa un conjunto de datos de Twitter para identificar emociones (usando un modelo basado en *BERT*).

Como novedad, en este artículo se revisaran modelos pre-entrenados con una mayor cantidad de información a los mostrados en el estado del arte y que no han sido usados en el análisis de polaridad de tuits, además se explora el reemplazo de palabras por sinónimos para mejorar los resultados en la tarea de análisis de polaridad de tuits.

3 Solución propuesta

Para la detección de la polaridad en tuits se proponen las siguientes etapas:

1. Pre-procesamiento del corpus.
2. Arquitectura de solución para detección de polaridad en tuits.
3. Programación y entrenamiento de redes neuronales de aprendizaje profundo.
4. Evaluación de los resultados.

A continuación se describe cada una de estas etapas.

3.1 Pre-procesamiento del corpus

Para el desarrollo, entrenamiento y evaluación del sistema se utilizará el corpus proporcionado en la página oficial del evento (*Workshop on Semantic Analysis at SEPLN 2020* [*TASS*], 2020*)*.

Como propuesta para este artículo se plantea separar los símbolos de las palabras, para posteriormente tokenizar e introducir cada tuit en el modelo pre-entrenado *BERT*. El objetivo es lograr que el modelo pre-entrenado identifique correctamente la mayor cantidad de palabras posible.

Otra parte importante en el pre-procesamiento es cambiar los nombres de usuario como "*@Erick*" y reemplazarlos por la palabra "usuario". Los hashtags como "*#10best*" también se reemplazan con la palabra "Tendencia" (como se hace en otras subtareas de análisis de polaridad (García J. et al., 2020)).

Otro elemento a revisar son los casos específicos dentro de un hashtag como "\#sarcasmo", donde la palabra "sarcasmo" indicaría que lo que está textualmente no está realmente expresado, como se menciona en el trabajo de Hernández (2021) para la detección de sarcasmo.

Uno de los principales problemas al utilizar el tokenizador de un modelo pre-entrenado es el hecho de que algunas palabras no se encontrarán dentro del vocabulario del modelo (debido a faltas de ortografía o regionalismos del español, palabras en otros idiomas, etc.), por esa razón, se propone detectar cuáles son las palabras que no están en el vocabulario y encontrar una palabra que tenga un significado similar que se encuentre dentro del vocabulario.

En la Figura 1 podemos observar que los tuits son pre-procesados como se ha explicado previamente, posteriormente pasan por la herramienta de tokenización de *BERT*, si la palabra que entra al tokenizador se encuentra en el vocabulario del modelo, se guarda con su respectivo valor numérico, de lo contrario, se busca un sinónimo de esa palabra que se encuentre en el vocabulario para reemplazarse (con un script se obtiene el sinónimo de la página de internet Wordreference (2022)).

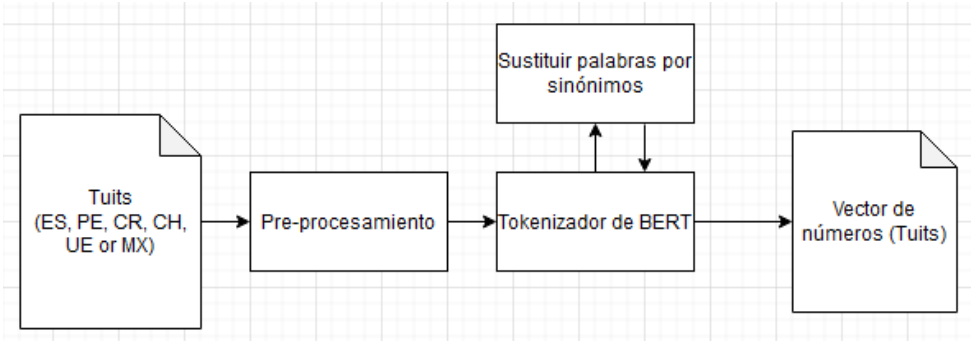


Fig. 1. Secuencia de pre-procesamiento de los tuits.

Respecto al etiquetado, se convirtió cada clase a un entero positivo, en este caso 0 para "N", 1 para "NEU" y 2 para "P".

3.2 Arquitectura de solución para detección de polaridad en tuits

Para esta etapa, se propone el uso de modelos pre-entrenados basados en *BERT* para detectar la polaridad de sentimientos en los tuits.

Para esta tarea se propone utilizar los siguientes modelos pre-entrenados basados en *BERT*: "*bert-base-multilingual-cased*" (Devlin et al., 2018) y "*IIC/beto-base-spanish-sqac*" (Gutiérrez et al., 2021).

"*Bert-base-multilingual-cased*" Es un modelo pre-entrenado en 104 lenguajes diferentes con información de *Wikipedia*, fue pre-entrenado con textos en minúsculas y tokenizados usando *WordPiece*, además tiene un tamaño de vocabulario compartido de 110,000 palabras.

"*IIC/beto-base-spanish-sqac*" Es un modelo pre-entrenado con un corpus masivo de 570GB de texto limpio y deduplicado con 135 mil millones de palabras extraídas de archivos web en español entre 2009 y 2019 (Gutiérrez et al., 2021) y posteriormente entrenado con el corpus *SQAC* (*Spanish Question-Answering Corpus*), este corpus contiene 6,247 contextos y 18,817 preguntas con sus respuestas, la información de origen pertenece a artículos enciclopédicos de *Wikipedia* en español, *Wikinoticias* en español, y textos del corpus español *AnCora*, que es una mezcla de diferentes fuentes de noticias y literatura. "*IIC/beto-base-spanish-sqac*" se basa en *RoBERTa-base (*que es una variación de *BERT)* y pertenece a una familia de modelos de lenguaje en español, estos modelos se pueden considerar unos de los modelos más grandes y mejores para el español (Gutiérrez et al., 2021).

En la Figura 2 se puede observar la arquitectura que se utilizará para especializar los modelos pre-entrenados *BERT* en la tarea de análisis de polaridad. Dentro de esta arquitectura se agrega una capa de abandono (*Dropout layer*) con una tasa de abandono de 0.5 y una capa que agrega una función de transformación lineal (*Linear layer*).

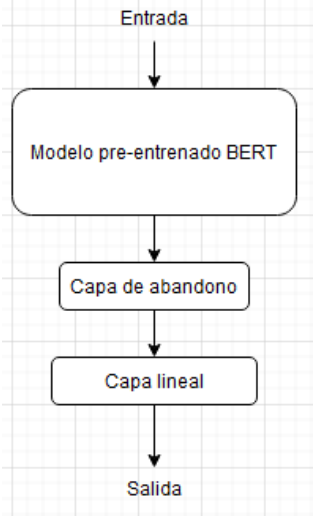


Fig. 2. Arquitectura para especialización de modelo pre-entrenado *BERT*.

3.3 Programación y entrenamiento de redes neuronales de aprendizaje profundo

Los modelos pre-entrenados (*BERT*) elegidos serán re-entrenados con el corpus pre-procesado para especializar el modelo en la detección de polaridad en cada tuit. Para ello se utilizarán librerías como *Pytorch (2021), Tensorflow (2021)* y *Keras (2021)* en *Python.*

El corpus de entrenamiento, estará conformado por el corpus de entrenamiento correspondiente de cada país.

Se utilizará el 90% del corpus de entrenamiento para entrenar el modelo, mientras que el 10% restante se utilizará para validación.

Cuando se encuentren los parámetros que funcionen mejor en el entrenamiento, se evaluará el modelo entrenado con el corpus de evaluación proporcionado por *TASS* 2020, para comparar los resultados con los de los participantes de *TASS* 2020.

3.4 Evaluación de los resultados

Los modelos pre-entrenados (*BERT*) elegidos serán re-entrenados con el corpus pre-procesado.

Para evaluar los resultados es necesario establecer los criterios de evaluación. Para la evaluación, se usarán las siguientes métricas: *Precisión*, *exhaustividad*, F_1 y Macro-F_1 (*macro average* F_1 *score*).

Los sistemas serán clasificados de acuerdo a la métrica Macro-F_1, esto significa que las medidas de *exhaustividad, precisión* y F_1 serán calculadas individualmente por cada clase ("P","N","NEU"), considerando cada variación de español (ES-España, PE-Perú, CR-Costa Rica, UR-Uruguay, MX-México) y posteriormente se obtendrá el promedio de todas las medidas F_1.

En la Ecuación 1 se define la métrica de la precisión, donde TP corresponde a las respuestas correctas y FP corresponde a las respuestas falsas.

En la Ecuación 2 se define la métrica de la exhaustividad, donde TP corresponde a las respuestas correctas y FN corresponde a las respuestas perdidas.

$$Precisión = \frac{TP}{TP + FP} \quad (1)$$

$$Exhaustividad = \frac{TP}{TP + FN} \quad (2)$$

$$F_1 = 2 * \frac{Precisión * Exhaustividad}{Precisión + Exhaustividad} \quad (3)$$

Finalmente, en la Ecuación 3 se define la métrica de F_1 que integra las métricas de precisión y la exhaustividad. Para la evaluación, se usará el archivo *gold* proporcionado en la página oficial del evento (*TASS*, 2020).

4 Resultados

A continuación se muestran los resultados encontrados en la revisión del corpus, la evaluación de los modelos usados y al final una comparación con los resultados de los participantes *TASS* 2020.

4.1 Descripción del corpus

Finalmente, en la Ecuación 3 se puede observar cómo obtener el F_1 en base a las métricas de precisión y la exhaustividad. Los conjuntos de entrenamiento y desarrollo incluyen una lista de tuits con su código correspondiente, el texto del tuit y su clasificación. El conjunto de prueba incluye una lista de tuits con el código del tuit y el texto. El conjunto de prueba incluye el código del tuit y su clasificación correcta.

Dentro del corpus, ningún tuit contiene más de 240 caracteres y contiene lenguaje informal (las faltas de ortografía, *emojis*, onomatopeyas son comunes).

La distribución de tuits por país y su clasificación se muestra en la Tabla 1 (para el corpus de entrenamiento).

Tabla 1. Distribución de tuits por país (corpus de entrenamiento)

Conjunto/Clasificación	**N**	**NEU**	**P**	**Total**
Costa Rica (CR)	475	297	354	1126
España (ES)	310	246	221	777
México (MX)	228	522	216	966
Perú (PE)	367	286	290	943
Uruguay (UY)	505	172	313	990

4.2 Resultados "*bert-base-multilingual-cased*"

En la Tabla 2 se puede ver la evaluación del modelo *'bert-base-multilingual-cased*', entrenado para cada variación del español, usando texto sin tokenizar y ningún tipo de pre-procesamiento.

Tabla 2. Evaluación del modelo "*bert-base-multilingual-cased*" sin pre-procesamiento.

Corpus	F_1 (N)	F_1 (NEU)	F_1 (P)	F_1 (Promedio)
ES	0.60742	0.66666	**0.69433**	0.65613
MX	**0.68192**	0.62415	0.68005	0.66204
PE	0.53063	**0.66912**	0.66009	0.61994
CR	0.60514	0.66309	**0.67334**	0.64719
UY	0.63769	0.65435	**0.67971**	0.65725
Promedio	0.61256	0.65547	**0.67750**	0.64851

En la Tabla 3 se puede ver la evaluación del modelo "*bert-base-multilingual-cased*", entrenado para cada variación del español, pre-procesando y tokenizando el texto.

Tabla 3. Evaluación del modelo "*bert-base-multilingual-cased*" con pre-procesamiento.

Corpus	F_1 (N)	F_1 (NEU)	F_1 (P)	F_1 (Promedio)
ES	0.75598	0.62877	**0.76793**	0.71756
MX	**0.80107**	0.53364	0.76642	0.70037
PE	0.72768	0.69387	**0.73666**	0.71940
CR	0.72684	0.65316	**0.72806**	0.70268
UY	0.74116	0.60194	**0.75645**	0.69985
Promedio	0.75054	0.62227	**0.75110**	0.70797

En la Tabla 4 se puede ver la evaluación del modelo "*bert-base-multilingual-cased*", entrenado para cada variación del español, pre-procesando, tokenizando el texto y reemplazando palabras no encontradas en el vocabulario por sinónimos que si se encontraran.

Tabla 4. Evaluación del modelo "*bert-base-multilingual-cased*" con reemplazo por sinónimos.

Corpus	F_1 (N)	F_1 (NEU)	F_1 (P)	F_1 (Promedio)
ES	0.75383	0.62006	**0.75959**	0.71116
MX	**0.78877**	0.49945	0.76642	0.68488
PE	0.72388	0.68686	**0.74358**	0.71810
CR	0.74032	0.65955	**0.74090**	0.71359
UY	0.74681	0.59330	**0.76260**	0.70090
Promedio	0.75072	0.61184	**0.75461**	0.70572

Se reemplazaron 363 palabras exitosamente, mientras que 47,230 palabras no se pudieron sustituir (de 47,593 palabras no encontradas en el vocabulario).

4.3 Resultados "*IIC/beto-base-spanish-sqac*"

En la Tabla 5 se puede ver la evaluación del modelo "*IIC/roberta-base-spanish-sqac*", entrenado para cada variación del español, usando texto sin tokenizar y ningún tipo de pre-procesamiento.

Tabla 5. Evaluación del modelo "*IIC/roberta-base-spanish-sqac*" sin pre-procesamiento.

Corpus	F_1 (N)	F_1 (NEU)	F_1 (P)	F_1 (Promedio)
ES	0.60742	0.66666	**0.69433**	0.65613
MX	0.61256	0.65547	**0.67750**	0.66204
PE	0.53063	**0.66912**	0.66009	0.61994
CR	0.60514	0.66309	**0.67334**	0.64719
UY	0.63769	0.65435	**0.67971**	0.65725
Promedio	0.61256	0.65547	**0.67750**	0.64851

En la Tabla 6 se puede ver la evaluación del modelo "*IIC/roberta-base-spanish-sqac*", entrenado para cada variación del español, pre-procesando y tokenizando el texto.

Tabla 6. Evaluación del modelo "*IIC/roberta-base-spanish-sqac*" con pre-procesamiento.

Corpus	F_1 (N)	F_1 (NEU)	F_1 (P)	F_1 (Promedio)
ES	0.79306	0.64183	**0.80378**	0.74622
MX	**0.81599**	0.50328	0.80030	0.70652
PE	**0.76923**	0.70375	0.76517	0.74605
CR	**0.78663**	0.67762	0.77241	0.74555
UY	**0.78663**	0.67762	0.77241	0.74555
Promedio	**0.79030**	0.64082	0.78281	0.73798

En la Tabla 7 se puede ver la evaluación del modelo "*IIC/roberta-base-spanish-sqac*", entrenado para cada variación del español, pre-procesando, tokenizando el texto y reemplazando palabras no encontradas en el vocabulario por sinónimos.

Tabla 7. Evaluación del modelo "*IIC/roberta-base-spanish-sqac*" con reemplazo por sinónimos.

Corpus	F_1 (N)	F_1 (NEU)	F_1 (P)	F_1 (Promedio)
ES	0.79448	0.64739	**0.80216**	0.74801
MX	**0.81599**	0.51627	0.79847	0.71024
PE	0.75899	0.70284	**0.76048**	0.74077
CR	**0.78933**	0.68959	0.77867	0.75253

UY	0.78006	0.63211	**0.80170**	0.73795
Promedio	0.78777	0.63764	**0.78829**	0.73790

Se reemplazaron 243 palabras exitosamente, mientras que 85,679 palabras no se pudieron sustituir (de 85,922 palabras no encontradas en el vocabulario).

4.4 Comparación resultados *TASS* 2020

En la Tabla 8 podemos observar los mejores resultados de los participantes de *TASS* 2020 en la tarea 1.1.

Tabla 8. Mejores resultados (F_1) de participantes de *TASS* 2020.

Corpus	***TASS* 2020 (F_1)**	**Participantes**
ES	0.61008	ELiRF-UPV
MX	0.63621	ELiRF-UPV
PE	0.60315	ELiRF-UPV
CR	0.61148	ELiRF-UPV
UY	0.62808	Palomino-Ochoa

A continuación en la Tabla 9 se puede observar la comparación de todas las propuestas implementadas (para "*bert-base-multilingual-cased*") con los mejores resultados de *TASS* 2020, como se observa, se ha logrado una mejora con el pre-procesamiento propuesto, y en algunos casos se mejora el F_1 con el reemplazo de sinónimos (posterior al pre-procesamiento).

Tabla 9. Comparación "*bert-base-multilingual-cased*" con mejores resultados (F_1) de *TASS* 2020.

Corpus	***TASS* 2020 (F_1)**	***BERT* (Multi) Sin pre-procesamiento (F_1)**	***BERT* (Multi) Con pre-procesamiento (F_1)**	***BERT* (Multi) Reemplazando sinónimos (F_1)**
ES	0.61008	0.65613	**0.71756**	0.71116
MX	0.63621	0.66204	**0.70037**	0.68488
PE	0.60315	0.61994	**0.71940**	0.71810
CR	0.61148	0.64719	0.70268	**0.71359**
UY	0.62808	0.65725	0.69985	**0.70090**

Tabla 10. Comparación "*IIC/roberta-base-spanish-sqac*" con mejores resultados (F_1) de *TASS* 2020.

Corpus	*TASS* 2020 (F_1)	*IIC/roberta* Sin pre-procesamiento (F_1)	*IIC/roberta* Con pre-procesamiento (F_1)	*IIC/roberta* Reemplazando sinónimos (F_1)
ES	0.61008	0.63109	0.74622	**0.74801**
MX	0.63621	0.66419	0.70652	**0.71024**
PE	0.60315	0.61011	**0.74605**	0.74077
CR	0.61148	0.63243	0.74555	**0.75253**
UY	0.62808	0.63873	**0.74555**	0.73795

En la Tabla 10 se puede observar la comparación de todos los sistemas implementados (con el modelo "*IIC/roberta-base-spanish-sqac*") con los mejores resultados de *TASS* 2020, al igual que el modelo anterior ("*bert-base-multilingual-cased*") se obtiene una mejora en el F_1 con el pre-procesamiento propuesto, y en algunos casos se mejora el F_1 con el reemplazo de sinónimos (posterior al pre-procesamiento).

Tabla 11. Comparación de los mejores resultados de cada modelo propuesto, con los mejores resultados (F_1) de *TASS* 2020.

Corpus	*TASS* 2020 (F_1)	*IIC/Roberta* (F_1)	*BERT* (Multi) (F_1)
ES	0.61008	**0.74801**	0.71756
MX	0.63621	**0.71024**	0.70037
PE	0.60315	**0.74605**	0.71940
CR	0.61148	**0.75253**	0.71359
UY	0.62808	**0.74555**	0.70090

Finalmente, en la Tabla 11 podemos ver una comparación de los mejores resultados obtenidos con cada modelo propuesto, el modelo "*IIC/roberta-base-spanish-sqac*" obtuvo los mejores resultados en estas pruebas, debido a que está especializado para textos en español, con "*bert-base-multilingual-cased*" se encontraron más palabras en su vocabulario, sin embargo, muchas de estas palabras no son necesariamente palabras en español (debido a que fue entrenado para múltiples idiomas). Por otra parte, la mejora que existe entre *TWilBERT* (el mejor modelo *BERT* en *TASS* 2020) y los modelos propuestos en este artículo, se debe a la cantidad y variedad de información de entrenamiento, siendo que *TWilBERT* fue entrenado principalmente con tuits y "*IIC/roberta-base-spanish-sqac*" fue pre-entrenado con un corpus masivo de 570GB (Gutiérrez et al., 2021).

Conclusiones

Con la finalidad de resolver el análisis de polaridad en tuits en español fueron implementados modelos pre-entrenados basados en BERT ("IIC/beto-base-spanish-sqac", "bert-base-multilingual-cased").

Los resultados obtenidos lograron superar la puntuación de los participantes de *TASS* 2020, esto se atribuye a que los modelos utilizados fueron entrenados con un conjunto de datos mucho más grande que el mejor modelo de los participantes (TWilBERT).

Se observó que "IIC/beto-base-spanish-sqac", obtuvo mejores resultados que "bert-base-multilingual-cased", debido a que la mayoría de las palabras encontradas en el vocabulario del modelo están dentro de un contexto en español.

El pre-procesamiento planteado da una mejora considerable en todas las pruebas de diferentes regiones y el reemplazo de sinónimos solo proporciona pequeñas mejoras en algunas regiones debido a que el sinónimo de una palabra no siempre es el mismo en diferentes regiones.

Como trabajo a futuro se propone mejorar el proceso de reemplazo de las palabras no encontradas, ya que la cantidad de palabras que se lograron reemplazar, fue inferior al 1% en ambos modelos, y los sinónimos no son los mismos en todas las regiones. También se propone comparar los resultados con otros modelos pre-entrenados como lo es "finiteautomata/beto-sentiment-analysis" (un modelo orientado específicamente al análisis de sentimientos en español, que no está orientado al análisis de tuits).

Referencias

Devlin, J., Wei Chang, M., Lee, K., y Toutanova, K. (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *CoRR*, vol. abs/1810.04805, pp. 1810.04805.

Fernández, D., (2021). "*El español: una lengua viva. Informe 2021*", *Instituto Cervantes*, pp. 5-6. Recuperado de https://cvc.cervantes.es/lengua/espanol_lengua_viva/pdf/espanol_lengua_viva_2021.pdf

García, J., Almela, A., Valencia, R., (2020). "UMUTeam at *TASS* 2020: Combining Linguistic Features and Machine-learning Models for Sentiment Classification" en García M., Gonzalo D., Martínez E., Martínez R., Rosso P., Jiménez S., Ortiz J., Miranda A., Porta J., Gutiérrez Y., Rosá A., Montes M., García M. (eds), *Proceedings172 of the Iberian Languages Evaluation Forum (IberLEF 2020) co-located with 36th Conference of the Spanish Society for Natural Language Processing (SEPLN 2020), Málaga, Spain, September 23th, 2020* (pp. 179-186), CEUR-WS.org.

García, M., Díaz, M., Plaza, M., Montejo, A., Jiménez, S., Martínez, E., Aguilar, A., Sobrevilla, M., Chiruzzo, L., Moctezuma, D., (2020). "Overview of *TASS* 2020: Introducing Emotion Detection" en García M., Gonzalo D., Martínez E., Martínez R., Rosso P., Jiménez S., Ortiz J., Miranda A., Porta J., Gutiérrez Y., Rosá A., Montes M., García M. (eds), *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020) co-located with 36th Conference of the Spanish Society for Natural Language Processing (SEPLN 2020), Málaga, Spain, September 23th, 2020* (pp. 163-170), CEUR-WS.org.

González, I. (2020). "Procesamiento del lenguaje natural con BERT: Análisis de sentimientos en tuits", *Universidad Carlos III de Madrid,* pp. 1-55.

González, J., Moncho, J., A., Hurtado, L., (2020). "ELiRF-UPV at *TASS* 2020: TWilBERT for Sentiment Analysis and Emotion Detection in Spanish Tweets" en García M., Gonzalo D., Martínez E., Martínez R., Rosso P., Jiménez S., Ortiz J., Miranda A., Porta J., Gutiérrez Y., Rosá A., Montes M., García M. (eds), *Proceedings of the Iberian Languages Evaluation Forum*

(IberLEF 2020) co-located with 36th Conference of the Spanish Society for Natural Language Processing (SEPLN 2020), Málaga, Spain, September 23th, 2020 (pp. 179-186), CEUR-WS.org.

Gutiérrez, A., Armengol, J., Pámies, M., Llop, J., Silveira, J., Carrino, P., Gonzalez, A., Armentano, C., Rodríguez, C., Villegas, M., (2021). "MarIA: Spanish Language Models", *CoRR*, vol. 68, pp. 39-60.

Hernández, J.C. (2021). "Aprendizaje Profundo y Neuroevolución para el Análisis de Sentimientos en Tweets Escritos en Español Mexicano", [Tesis de maestria, *Universidad Veracruzana]. https://www.uv.mx/personal/emezura/files/2022/01/Tesis-Clemente.pdf*

Keras, (2021). "*Getting started*". Recuperado de https://keras.io/getting_started/

Mazo, J.D. (2021). "Detección de palabras clave en el análisis de sentimiento de tweets usando técnicas de ML", [Trabajo de grado especialización, *Universidad de Antioquia]. https://bibliotecadigital.udea.edu.co/bitstream/10495/24326/3/MazoJulian_2021_PalabrasDeteccionSentimiento.pdf*

Palomino, D., Ochoa, J., (2020). "Palomino-Ochoa at *TASS* 2020: Transformer-based Data Augmentation for Overcoming Few-Shot Learning" en García M., Gonzalo D., Martínez E., Martínez R., Rosso P., Jiménez S., Ortiz J., Miranda A., Porta J., Gutiérrez Y., Rosá A., Montes M., García M. (eds), *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020) co-located with 36th Conference of the Spanish Society for Natural Language Processing (SEPLN 2020), Málaga, Spain, September 23th, 2020* (pp. 171-178), CEUR-WS.org.

Pérez, J., Manuel, Giudici, Carlos, y Luque, F. (2021). "Pysentimiento: A Python Toolkit for Sentiment Analysis and SocialNLP tasks", *arXiv*, vol. 1, pp. 1-4.

pytorch, (2021). "*End-to-end Machine Learning Framework*", Recuperado de https://pytorch.org/features/

Scola, E. y Segura, I. (2021). "Sarcasm Detection with BERT", *Procesamiento del Lenguaje Natural*, vol. 67, pp. 13-25.

Workshop on Semantic Analysis at SEPLN, (2020). "*About*". Recuperado de http://tass.sepln.org/2020/

TensorFlow, (2021). "*Por qué TensorFlow*". Recuperado de https://www.tensorflow.org/?hl=es-419

López, C., Gonzales, S., Orlando (2021). "Análisis de sentimiento de comentarios en español en Google Play Store usando BERT", *scielocl*, vol. 29, pp. 557 - 563.

Valladares, J.G. (2022). "Análisis de sentimientos para textos cortos en español, una revisión del estado del arte", *Universidda Politécnica Salesiana Sede Quito,* pp. 4-14.

Wordreference, (2022). "Diccionario de sinónimos y antónimos", Recuperado de https://www.wordreference.com/sinonimos/

Zárate, G.H. (2021). "Análisis de sentimientos en información de medios periodísticos y redes sociales mediante redes neuronales recurrentes", [Tesis para obtener el título profesional de Ingeniero, *Pontificia Universidad Católica del Perú]*. https://tesis.pucp.edu.pe/repositorio/bitstream/handle/20.500.12404/21525/ZARATE_CALDERON_GABRIEL_ANALISIS_SENTIMIENTO_INFORMACION.pdf

Capítulo 4

Descubrimiento de tópicos a partir de textos científicos en español

Josué Padilla-Cuevas[1], Gabriela A. García-Robledo[1], José A. Reyes-Ortiz[1]

[1] Universidad Autónoma Metropolitana Unidad Azcapotzalco
Departamento de Sistemas
{jpc,gagr,jaro}@azc.uam.mx

Resumen. Los artículos científicos son documentos que contienen información valiosa sobre una gran cantidad de temas. En el dominio de la salud, grandes cantidades de textos científicos son generados diariamente, los cuales hablan de temas como vacunaciones, enfermedades, tratamientos y estadísticas. Esta información se encuentra en los artículos científicos sin una organización temática específica, lo que complica a los lectores y consumidores de esta información cuando deciden analizar, clasificar u organizarlos por algún tema o tópico. Por ello, surge la necesidad de contar con sistemas, algoritmos o métodos de análisis automático de este tipo de textos para encontrar los tópicos mencionados en ellos. De esta manera, en este artículo se presenta un enfoque para el descubrimiento de tópicos a partir de artículos científicos en español sobre el domino clínico, utilizando el algoritmo LDA (de su nombre en inglés *Latent Dirichlet Allocation*) ampliamente utilizado en la literatura para esta tarea. Un proceso de evaluación basada en la coherencia de los tópicos es presentado, logrando resultados de coherencia de 0.7189.

Palabras clave: Descubrimiento de tópicos, textos científicos español, textos clínicos, algoritmo LDA.

1. Introducción

El descubrimiento de tópicos se define como la tarea de encontrar los tópicos relevantes presentes en un conjunto de documentos como entrada. Esta tarea puede ser aplicada a cualquier tipo de textos como los artículos científicos. En el dominio clínico se produce, diariamente, una gran cantidad de documentos científicos que necesitan ser organizados por temas o que los usuarios de ellos requieren conocer los temas plasmados en ellos. Estos usuarios invierten una gran cantidad de tiempo y esfuerzo si esta tarea la realizaran de manera manual. En ese sentido hay algoritmos o programas computacionales que pueden realizar el descubrimiento de tópicos de manera automática sobre una gran cantidad de documentos. Por ello, en este trabajo se presenta un enfoque para el descubrimiento de tópicos a partir de un conjunto de 234,547 artículos científicos en español del domino clínico. En este enfoque se utiliza el algoritmo LDA y las palabras procesadas de los documentos para descubrir los tópicos *T* con sus respectivas *top-k* palabras

representativas y la distribución de cada documento en los tópicos. Además, se introduce un proceso de evaluación basada en la coherencia global post-normalizada de los tópicos basada en la información puntual mutua de las *top-10* palabras que conforman cada tópico. Esta información puntual mutua es obtenida con un conjunto externo de documentos, que en nuestro caso es la Wikipedia en español compuesta por arriba de 4 millones de artículos.

El resto del artículo se organiza como sigue. La Sección 2 presenta el estado del arte relacionado con el descubrimiento automático de tópicos a partir de textos del dominio clínico en cualquier idioma, incluido el español, italiano, francés e inglés. En la Sección 3, se presenta el enfoque propuesto para el descubrimiento de tópicos en textos de artículos científicos en español del dominio clínico. En la Sección 4 se presentan los experimentos realizados y los resultados obtenidos. Finalmente, las conclusiones incluyendo el trabajo a futuro son presentados en la Sección 5.

2. Estado del arte

En esta sección se presentan artículos relacionados con el descubrimiento de tópicos a partir de conjuntos de textos científicos para el dominio clínico, presentando, discutiendo y analizando el uso de algoritmos, métodos, enfoques y diversas técnicas para este proceso.

Los textos científicos (artículos de investigación) ofrecen información certera sobre diversas temáticas, como en el caso del dominio de la salud, de diversas enfermedades, pandemias, tratamientos, medicamentos, por mencionar algunos. En este sentido se han propuesto diversos trabajos para el análisis temático de textos científicos, en los cuales el uso del algoritmo LDA ha sido el más utilizado en la literatura: (De Santis et al, 2020), (Yu et al, 2020), (Ta et al, 2020), (Vijayan, 2021), (Chen et al, 2019), (Abuhay et al, 2018), (Yau et al, 2014).

El seguimiento y análisis temático de la pandemia originada por el virus SARS-CoV-2 utilizando las redes sociales y datos de internet como fuentes de información, ha sido abordado en los últimos años en (De Santis et al, 2020), donde se realiza un seguimiento y rastreo de tópicos relevantes durante la pandemia del COVID-19 en Italia; mostrando un proceso de evaluación mediante gráficas de los tópicos descubiertos; también en (Yu et al, 2020), donde se identifican los tópicos presentes durante la pandemia por COVID-19 a partir de los titulares de los diarios españoles El país y El mundo. En (Ta et al, 2020) se presenta una minería de tópicos, donde se realiza un filtrado de textos del diario Al-Jazeera originario de Qatar que contengan la palabra COVID y a partir de ellos se descubren tópicos mediante el algoritmo LDA.

Por su parte, en India se realizó una investigación mostrada en (Debnath et al, 2020) sobre la formación de políticas reactivas para combatir el coronavirus en distintos sectores públicos. Los datos se obtienen de la Oficina de Información de Prensa (PIB) los cuales involucran 260,852 palabras a partir de 396 documentos y se realiza una extracción de temas de alta probabilidad utilizando el algoritmo LDA para determinar el aprovechamiento de las normas de bloqueo y distanciamiento social creadas por el primer ministro de la India.

El modelo de clasificación de la desinformación de COVID-19 y descubrimiento de temas llamado CANTM de (Song et al, 2021) es creado para ser un auxiliar en la transmisión de mensajes de salud pública efectivos y combatir la desinformación entre la comunidad. El corpus utilizado es un conjunto de datos clasificados de acuerdo con el tema de afirmaciones falsas sobre el COVID-19 publicado en el sitio web de International Farm Comparison Network (IFCN).

Por otro lado, existen trabajos que utilizan la literatura científica para el descubrimiento de tópicos. En (Cao et al, 2022) se realiza un estudio comparativo de modelado de tópicos a partir de resúmenes o textos completos de la literatura investigadora con temas del COVID-19 utilizando el algoritmo LDA en el data set CORD-19. Para realizar una evaluación de la influencia de los distintos tipos de documentos utilizados para el modelado de tópicos se utilizan las etapas de preprocesamiento, modelado y análisis de tópicos. En (Vijayan, 2021) se modela temáticamente la literatura de enseñanza y aprendizaje sobre el COVID-19. Utiliza datos extraídos de *Scopus* y la identificación de temas clave sobre cada investigación. Los tópicos son extraídos de los resúmenes implementando el algoritmo LDA en la plataforma KNIME (*Konstanz Information Miner*). El sistema descrito en (Chen et al, 2019) realiza un análisis de tópicos en la literatura científica a partir de los resúmenes de publicaciones de células solares sensibilizadas por colorante obtenidas de búsquedas en *World of Science* aplicando el algoritmo LDA. Además, se muestra un enfoque de identificación de relaciones en los tópicos encontrados para obtener una red temática de toda la información. El método de (Abuhay et al, 2018) realiza la predicción de la tendencia en los temas de investigación. Se utilizan artículos de la Conferencia Internacional sobre Ciencias Computacionales junto con el método de modelado de temas de factorización matricial no negativa para el descubrimiento de temas. También, la investigación de los métodos de modelado de tópicos, tal como LDA y sus extensiones a partir de publicaciones científicas de varios grupos se muestra en (Yau et al, 2014), en la evaluación de resultados se utilizaron documentos de distintos campos para determinar si su agrupación fue correcta.

En (Cinelli et al, 2020) también se realiza un análisis de la información sobre el COVID-19 en redes sociales y fuentes cuestionables identificando la información errónea. Se muestran estimaciones numéricas de los rumores en cada plataforma. Para la evaluación de los temas relacionados con el coronavirus se utiliza el algoritmo *Partitioning Around Medoids* (PAM) en su representación vectorial y para la construcción de incrustaciones de palabras en cada corpus de redes sociales se utiliza el algoritmo *Skip-gram*.

El estudio de (Älgå et al, 2020) explora la cantidad de investigaciones científicas publicadas sobre COVID-19 y su evolución durante la fase inicial de la pandemia. La búsqueda de títulos, palabras clase y resúmenes fue realizada en *PubMed*. Se utiliza el algoritmo LDA en la extracción de los temas que fueron analizados para determinar la tendencia según los cambios temporales en la investigación de cada tema, impacto de revista u origen geográfico.

También, se han presentado trabajos para el análisis de tópicos sobre las enfermedades con más incidencias en el mundo. Es el caso del trabajo presentado en (Ghosh et al, 2013) donde se analizan los tópicos presentes en los mensajes de Twitter de los Estados Unidos de América, relacionado con *Childhood*, *Obesity*, *McDonalds*, mostrando como resultado gráficas de barras con los lugares (geolocalizaciones de los tweets) relacionados con los tópicos de interés; el análisis de tópicos utilizando el

algoritmo LDA a partir de textos científicos de enfermedades ha sido abordado para el cáncer (Kumar y Greiner, 2019) y diabetes (Harris et al, 2015).

En (Huang et al, 2014) se propone un enfoque para el descubrimiento de patrones en protocolos de atención médica a partir de registros de eventos clínicos utilizando el algoritmo LDA junto con un modelado especializado para el descubrimiento de patrones. En (Kayi et al, 2013) se desarrolla un sistema clasificador basado en temas a partir de registros de salud electrónicos. Este sistema comienza con una etapa de preprocesamiento de la información médica protegida, continua con la implementación del algoritmo LDA para generar los modelos temáticos y un modelado de temas de informes, utiliza el algoritmo SVM (*Support vector machine*) para demostrar el buen desempeño de la clasificación, y realiza la creación de un clasificador de temas agregado y la clasificación binaria de tópicos con la premisa de que cada tema corresponde a solo una clase. La plataforma PubMed también es utilizada en (van Altena et al, 2016) donde se realiza una investigación del término grandes volúmenes de datos para su mejor comprensión, aplicando un enfoque sistemático basado en literatura científica biomédica. Los autores parten de cuatro temas clave identificados por un trabajo anterior y se añaden ocho temas sobre definiciones existentes del término "grandes volúmenes de datos".

Como se puede observar en la revisión del estado del arte, la mayoría de trabajos se enfoca en el algoritmo LDA para textos clínicos. Por ello, este trabajo tiene como objetivo presentar un descubrimiento de tópicos a partir de un conjunto de textos clínicos (artículos científicos) en español utilizando el algoritmo. Además, es de observarse que el proceso de evaluación de la tarea de descubrimiento de tópicos es escaso, por ello este artículo presenta una evaluación basada en la coherencia de los tópicos descubiertos utilizando un conjunto de más de 4 millones de textos de la Wikipedia en español como conjunto de textos externos.

3. Enfoque propuesto para el descubrimiento de tópicos

En esta sección se presenta la propuesta de solución de este trabajo para el descubrimiento de tópicos a partir de un conjunto de textos clínicos en español y utilizando un conjunto de textos externos (Wikipedia en español) para el proceso de evaluación de la coherencia de los tópicos. La Figura 1 muestra esta arquitectura donde el proceso completo involucra como entrada ambos conjuntos de textos en español; un módulo de preprocesamiento de textos, la tarea de descubrimiento de tópicos mediante el algoritmo LDA; los conjuntos de tópicos descubiertos y la evaluación de esta tarea mediante coherencia de los tópicos.

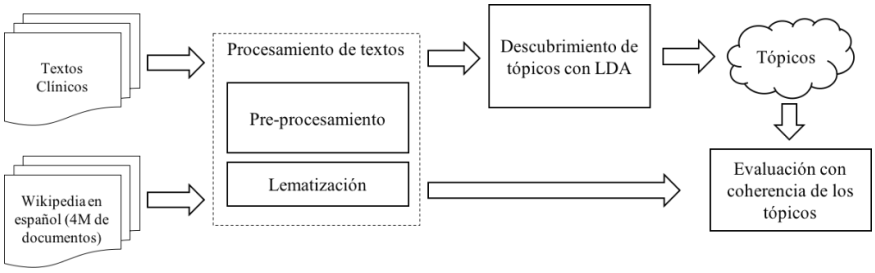


Figura 1. Arquitectura de solución para el descubrimiento de tópicos

3.1. Conjunto de datos

Los conjuntos de textos en español utilizados consisten en dos partes, los cuales son utilizados a lo largo del proceso de descubrimiento de tópicos y se describen a continuación.

Conjunto de textos clínicos. Este conjunto consta de 234,547 documentos clínicos en español que fueron obtenidos de la competencia BioASQ Workshop 2021[1]. Estos documentos se encuentran en formato JSON y son utilizados para el descubrir los tópicos presentes mediante el algoritmo LDA. A pesar de que el conjunto de documentos proporcionado en esta tarea está dedicado para sistemas de pregunta-respuesta en el dominio clínico, ellos son utilizados en este artículo para descubrir tópicos desde cero por su diversidad de tópicos dentro de la medicina, por mencionar algunos: oncología, diabetes, COVID e hipertensión. Además, el descubrimiento de tópicos se obtiene desde cero por lo que no se requiere de un conjunto de datos previamente etiquetado con tópicos y su evaluación está basada en un conjunto de textos externos, en este caso Wikipedia en español. Un ejemplo de estos textos se muestra en la Figura 2.


```
{"articles":[
{"id": "ibc-ET6-1764", "title": "La Quercetina como un potencial
nutracéutico contra la enfermedad por coronavirus 2019 (COVID-19)",
"abstractText": "INTRODUCCIÓN: La enfermedad del coronavirus 2019
(COVID-19) es una enfermedad viral que afecta a varios órganosy sistemas.
Los tratamientos preventivos o profilácticos son especialmente útiles en
enfermedades infecciosas emergentes como COVID-19 porque reducen la
necesidad de hospitalización y el gasto en salud pública. Aunque el efecto
preventivo del SARS-CoV-2 de varios agentes terapéuticos (e.g.,
hidroxicloroquina/cloroquina, remdesivir,lopinavir y ritonavir) se ha
evaluado ampliamente, ninguno de ellos ha demostrado una gran eficacia
clínica. MÉTODO: Por lo tanto, aquí nuestro objetivo es abordar y discutir
los estudios publicados recientemente sobre el potencial quimioprofilático
de la quercetina contra el SARS-CoV-2. METODOLOGÍA: Se realizó una
búsqueda de la literaturaen bases como PubMed/MEDLINE, Scielo, Scorpus,
Web of Science, Cochrane Library y Clinical Trials.gov. Se incluyeron y
evaluaron críticamente estudios que abordan la quercetina contra el SARS-
CoV-2 u otros tipos decoronavirus. RESULTADOS: Algunos estudios han
demostrado que la quercetina, un flavonoide aprobado por la FDA que se
utiliza como agente antioxidante y antiinflamatorio, inhibe la entrada del
coronavirus (SARS-CoV) en la célula huésped.Además, un estudio in silico
mostró que la quercetina es un potente inhibidor de la proteasa principal
del SARSCoV-2 (Mpro), lo que sugiere que este flavonoide también es activo
contra COVID-19. CONCLUSIONES: Debido a que la quercetina podría prevenir
y disminuir la duración de las infecciones por SARSCoV-2, es plausible
suponer que el uso profiláctico de este flavonoide produce varios
beneficios clínicos. Pero, estas pruebas preliminares deben ser
confirmadas mediante ensayos in vitro y, posteriormente, en un ensayo
clínico aleatorizado", "journal": "Ars pharm", "year": 2021, "db":
"IBECS", "decsCodes": ["D006801", "D018352", "D015203", "D019587",
"D065129", "D000998", "D011480", "D011794"]},
```


Figura 2. Ejemplo de textos clínicos en español

[1] http://www.bioasq.org/workshop2021

Conjunto de Wikipedia en español. En este trabajo se utiliza un conjunto de artículos de la Wikipedia en español[2] como recurso textual externo para la etapa de evaluación mediante la métrica de coherencia de los tópicos. Este conjunto consta de 4,236,176 artículos en formato XML y un ejemplo de ellos se muestra en la Figura 3.

```
  <page>
    <title>Artes visuales</title>
    <ns>0</ns>
    <id>19</id>
    <revision>
      <id>142881247</id>
      <parentid>142881246</parentid>
      <timestamp>2022-04-13T15:51:21Z</timestamp>
      <contributor>
        <username>SeroBOT</username>
        <id>4980693</id>
      </contributor>
      <minor />
      <comment>Revertidos los cambios de [[Special:Contributions/190.239.238.171|190.239.238.171]] ([[User
talk:190.239.238.171|disc.]]) a la última edición de SeroBOT</comment>
      <model>wikitext</model>
      <format>text/x-wiki</format>
      <text bytes="6002" xml:space="preserve">{{redirige aquí|Artes Visuales|Diseño|actividad artística de la
industrialización}}
[[Archivo:Mona Lisa, by Leonardo da Vinci, from C2RMF retouched.jpg|thumb|right|220px|La [[Mona Lisa]] es uno de los
cuadros más reconocidos de [[Occidente]].]]
[[Archivo:Wolf Vostell, Fiebre de Automóvil, 1973, Instalación.JPG|thumb|220 px|[[Wolf Vostell]], ''Fiebre de
Automóvil'' (1973), [[Instalación artística|instalación]], [[Museo Vostell-Malpartida]].]]
[[Archivo:artistIsPresent.jpg|thumb|220 px|[[Marina Abramović]], ''La artista está presente'' (2010), [[Performance]],
[[Museo de Arte Moderno de Nueva York|MoMa]].]]
[[Archivo:Christo Rifle Gap.jpeg|thumb|220 px|[[Christo y Jeanne-Claude]], ''Valley Curtain'' (1971), [[Instalación
artística|Instalación]] ([[Land Art]]), Rifle Gap, [[Montañas Rocosas]], [[Colorado]].]]

Las '''artes visuales''' engloban las [[artes plásticas]] tradicionales, así como las expresiones que incorporan a la
nueva tecnología orientada al arte o elementos no convencionales, y que su mayor componente expresivo es visual, como la
[[fotografía]], [[video|videografía]], [[cinematografía]], y lo también llamado [[arte de los nuevos medios]], entre los
que se incluyen:
* [[arte digital]]
* [[fanart]]
* [[fotografía]]
* [[net.art]]
* [[videoarte]]
```

Figura 3. Ejemplo de un texto de la Wikipedia en español.

3.2. Procesamiento de los textos

En este trabajo se realiza un procesamiento de los textos con la finalidad de prepararlo para la identificación de tópicos con una mejor precisión. Esta etapa consta de diversas tareas que son aplicadas a ambos conjuntos de textos en español según corresponda, las cuales se describen a continuación. Estas tareas fueron implementadas Python.

Extracción. Los textos son extraídos de los archivos en formato JSON y XML mediante el análisis de etiquetas. El resultado de esta tarea son textos planos sin etiquetas ni marcas para su posterior procesamiento.

Limpieza. Los caracteres especiales, números y URL son eliminadas de los textos.

Eliminar palabras vacías. Los textos contienen palabras que no aportan un significado a los tópicos y el objetivo de esta tarea es eliminarlas para mejorar la coherencia de los tópicos descubiertos. En estas palabras se encuentran, por mencionar algunas, las preposiciones (*a, ante, para, por*) y los artículos (*el, los, las, unos*).

Lematización. Esta tarea consiste en llevar las palabras a su raíz léxica con la finalidad de normalizar y unificar las diferentes flexiones o conjugaciones de las palabras. El objetivo de esta tarea es reducir el tamaño del vocabulario de los documentos para obtener tópicos representativos. Por mencionar algunas reglas aplicadas aquí son,

[2] https://es.wikipedia.org/wiki/Wikipedia:Descargas

los verbos son convertidos a su forma infinitiva (*corrieron->correr*, *correrán->correr*); los sustantivos son trasladados a su forma singular masculino (*niñas->niño*).

Conversión a minúsculas. Los textos son convertidos a minúsculas con la finalidad de homogenizarlos.

3.3. Descubrimiento de tópicos

El proceso de descubrimiento de tópicos consiste en encontrar de manera automática los tópicos relevantes existentes en un conjunto de documentos de entrada. En este trabajo se utiliza el conjunto de 234,547 textos científicos en español descrito anteriormente para esta etapa de descubrimiento de tópicos utilizando el algoritmo LDA (por sus siglas en ingles de *Latent Dirichlet Allocation*).

El algoritmo LDA es uno de los algoritmos tradicionales más popular para el descubrimiento de tópicos descubriendo las estructuras semánticas a partir de un conjunto de textos (D. M. Blei et al, 2003), esta popularidad quedó demostrada con la revisión del estado el arte (Sección 2). Este algoritmo captura patrones de co-ocurrencia de las palabras a nivel de documento. Esto significa que entre más co-correncias tenga un tópico descubierto mayor será de confiabilidad.

En este artículo se utilizan hiper-parámetros $\alpha = 0.05$ y $\beta = 0.01$ para el algoritmo LDA tomados de (Qiang et al, 2020), quienes mencionan se pueden aplicar en textos de cualquier longitud. El proceso se puede definir como, dado un conjunto de textos (D) con un vocabulario (V) y un número predefinido de tópicos (T), se descubren mediante el algoritmo LDA:

(1) Un conjunto tópicos($t_i \in T$) con sus *top-k* palabras (K).
(2) La distribución de cada documento en cada uno de los tópicos.

La implementación de este algoritmo se ha realizado utilizando la librería Java de código abierto denominada STTM y proporcionada en (Qiang et al, 2020), la cual incluye la implementación del algoritmo LDA y permite realizar experimentos con diversos valores del número de tópicos (T).

3.4. Métrica de evaluación de la tarea

La evaluación de la tarea de descubrimiento de tópicos está aún en investigación y diversas métricas se han propuesto recientemente. Por ello, en este artículo se utilizan las propuestas de (Qiang et al, 2020) y (Quan et al, 2015) sobre la métrica denominada Coherencia de los Tópicos. Estas propuestas se adaptan para proponer la Coherencia de Tópicos Normalizada. Primero se obtiene la Coherencia Normalizada (*CohN*) de cada Tópico (t_i) de la forma que se presenta en la Ecuación (1).

$$CohN(t_i) = \frac{2}{K(K-1)} \sum_{j=2}^{K} \sum_{i=1}^{j-1} \frac{log \frac{P(w_i,w_j)}{P(w_i)P(w_j)}}{-\log P(w_i,w_j)} \quad (1)$$

Esta Coherencia Normalizada se basa en obtener la información puntual mutua normalizada (NPMI por sus siglas en inglés) de cada par de palabras que pertenecen al

top-k de palabras de cada tópico. $P(w_i,w_j)$ es la probabilidades de que las palabras w_i y w_j co-ocurran en un mismo párrafo dentro del conjunto de textos externos, en nuestro caso la Wikipedia en español. $P(w_i)$ y $P(w_j)$ es la probabilidad de que la palabra w_i y w_j, respectivamente, co-ocurran en un mismo párrafo.

Después, en este artículo se propone obtener la Coherencia Global Normalizada (*CohGloN*) como un promedio de la Coherencia Normalizada de todos los tópicos (t_i), mediante la Ecuación (2).

$$CohGloN = \frac{1}{T} \sum_{i=1}^{T} CohN(t_i) \quad (2)$$

La Coherencia Global Normalizada proporciona valores del rango [-1,1], por lo que se aplica una normalización posterior mediante la Ecuación (3). La finalidad de esta post-normalización es obtener valores del rango [0,1] para la mejor compresión de los resultados. X es el valor a normalizar; y para nuestro caso X_{min} equivale a -1 y X_{max} equivale a 1.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (3)$$

4. Resultados experimentales

El algoritmo LDA es aplicado sobre los 234,547 artículos científicos en español del dominio clínicos para obtener los tópicos, esta tarea es evaluada con la métrica de coherencia de los tópicos. Con la finalidad de realizar una evaluación detallada, se llevaron a cabo diversas ejecuciones del algoritmo modificando el valor del número de tópicos (T) con los siguientes valores de entrada: 5, 10, 20, 30, 50, 80, 100, 150, 200, 250, 300, 400, 500. El valor utilizado en todos los experimentos para el número de palabras clave (K) es de 10, por lo tanto se obtienen el *top-10* de palabras representativas para cada tópico descubierto.

Los resultados de la Coherencia Global post-normalizada considerando el conjunto de textos externos (Wikipedia en español de 4 millones de artículos) para cada valor de tópico se muestra en la Tabla 1.

Tabla 1. Resultados de coherencia de los tópicos

T	Coherencia Global post-normalizada
5	0.6755
10	0.6790
20	0.6830
30	0.6967
50	0.7011
80	0.7063
100	0.7107
150	**0.7189**

200	0.7034
250	0.6950
300	0.6634
400	0.5912
500	0.5780

La Tabla 1 muestra que el valor de 150 tópicos proporciona el mejor valor de coherencia de tus palabras *top-10* descubiertas considerando una coherencia global de 0.7189 obtenida del promedio de la coherencia de todos los tópicos. En la Tabla 2 se muestran, a manera de ejemplo, algunas palabras del *top-10* de algunos de los 150 tópicos descubiertos.

Tabla 2. Ejemplos de tópicos descubiertos con el valor de *T=150* mediante al algoritmo LDA

Número del tópico	**Palabras del top-10 del tópico**
16	**cáncer**, quimioterapias, pecho, pulmón, diagnóstico, biopsia, oncológico, …
57	vacuna, virus, **covid**, china, **pandemia**, sars-cov, casos, ...
93	**diabetes**, **obesidad**, comida, dieta, insulina, mellitus, …
123	**cardiovascular**, **corazón**, infarto, arteria, vascular, miocardio, …

Con los resultados de la Tabla 2 se puede identificar que el tópico 57 pertenece a la enfermedad COVID-19 que ha causado una pandemia en los últimos años, un hallazgo significado al descubrir un tópico, entre otras cosas, relacionado con vacunas para el COVID. Además, se obtienen los documentos distribuidos en este tópico. El objetivo de realizar una experimentación modificando el valor del número de tópicos fue para encontrar los temas contenidos en el conjunto de documentos clínicos a partir de cero.

5. Conclusiones y trabajo a futuro

En el presente trabajo se ha presentado un enfoque para el descubrimiento de tópicos a partir de textos de artículos científicos en español del dominio clínico, utilizando el algoritmo LDA. El proceso completo ha consistido en partir del conjunto de textos clínicos, aplicarles una etapa de procesamiento, el uso del algoritmo LDA sobre dichos textos para descubrir tópicos y una evaluación mediante la coherencia de los tópicos descubiertos.

Las principales aportaciones de este artículo son: a) la creación de dos conjuntos de textos (uno para descubrir los tópicos y otro utilizado como recurso externo para la evaluación); b) la aplicación de tareas de Procesamiento de Lenguaje Natural(PLN) sobre los textos, entre las que destacan la limpieza y la lematización utilizada para reducir el vocabulario de los textos; c) el uso del algoritmo LDA para textos en español; d) la propuesta de evaluación con la métrica de coherencia post-normalizada de los tópicos;

e) el resultado obtenido de coherencia 0.7189 para el valor de 150 tópicos para el conjunto de textos clínicos.

El objetivo de llevar a cabo una experimentación con diversos valores de *T* fue obtener el número de tópicos con mejor coherencia, logrando una coherencia global post-normalizada de 0.7189 para 150 tópicos. Las palabras clave de los tópicos descubiertos pueden ser de gran utilidad para los usuarios que requieren analizar los conjuntos de artículos científicos y encontrar las temáticas referidas y la distribución temática de ellos.

Como trabajo a futuro se propone el uso de otros algoritmos para el descubrimiento de tópicos como el BTM (*Biterm Topic Model* por sus siglas en inglés) o PLSA (por sus siglas en inglés de *Probabilistic Latent Semantic Analysis*), así como la aplicación de técnicas de PLN como es el uso de frases nominales en lugar de simples palabras para su posterior descubrimiento de tópicos a nivel de frases de los textos.

6. Referencias

De Santis, E., Martino, A., y Rizzi, A. (2020). "An Infoveillance System for Detecting and Tracking Relevant Topics From Italian Tweets During the COVID-19 Event", *IEEE Access,* vol. 8, pp. 132527–132538.

Yu, J., Lu, Y., y Muñoz-Justicia, J. (2020). "Analyzing Spanish News Frames on Twitter during COVID-19—A Network Study of El País and El Mundo", *International Journal of Environmental Research and Public Health*, vol. 17, pp. 5414.

Ta, T. H., Rahman, A. B. S., Sidorov, G., y Gelbukh, A. (2020). "Mining Hidden Topics from Newspaper Quotations: The COVID-19 Pandemic", *Advances in Computational Intelligence,* pp. 51–64.

Debnath, R., y Bardhan, R. (2020). "India nudges to contain COVID-19 pandemic: A reactive public policy analysis using machine-learning based topic modelling", *PLoS One*, vol. *15*, e0238972.

Song, X., Petrak, J., Jiang, Y., Singh, I., Maynard, D., y Bontcheva, K. (2021). "Classification aware neural topic model for COVID-19 disinformation categorization", *PloS one*, vol. 16, e0247086.

Cao, Q., Cheng, X., y Liao, S. (2022). "A comparison study of topic modeling based literature analysis by using full texts and abstracts of scientific articles: a case of COVID-19 research", *Library Hi Tech*, (ahead-of-print).

Vijayan, R. (2021). "Teaching and learning during the COVID-19 pandemic: A topic modeling study", *Education Sciences*, vol. 11, pp. 347.

Chen, H., Wang, X., Pan, S., y Xiong, F. (2019). "Identify topic relations in scientific literature using topic modeling", *IEEE Transactions on Engineering Management*, vol. 68, pp. 1232-1244.

Abuhay, T. M., Nigatie, Y. G., y Kovalchuk, S. V. (2018). "Towards predicting trend of scientific research topics using topic modeling", *Procedia Computer Science*, vol. 136, pp. 304-310.

Yau, C. K., Porter, A., Newman, N., y Suominen, A. (2014). "Clustering scientific documents with topic modeling", *Scientometrics,* vol. 100, pp. 767-786.

Cinelli, M., Quattrociocchi, W., Galeazzi, A., Valensise, C. M., Brugnoli, E., Schmidt, A. L., ... y Scala, A. (2020). "The COVID-19 social media infodemic", *Scientific reports*, vol. 10, pp. 1-10.

Älgå, A., Eriksson, O., y Nordberg, M. (2020). “Analysis of scientific publications during the early phase of the COVID-19 pandemic: topic modeling study”, *Journal of medical Internet research*, vol. 22, e21559.

Ghosh, D. D., y Guha, R. (2013). “What are we ‘tweeting’ about obesity? Mapping tweets with topic modeling and Geographic Information System”, *Cartography and Geographic Information Science*, vol. 40, pp. 90–102.

Kumar, Luke y Greiner, Russ. (2019). “Gene expression based survival prediction for cancer patients—A topic modeling approach”, *PLOS ONE*, vol. 14, e0224446.

Harris, J. K., Mart, A., Moreland-Russell, S., y Caburnay, C. A. (2015). “Peer Reviewed: Diabetes topics associated with engagement on twitter”, *Preventing chronic disease*, vol. 12.

Huang, Z., Dong, W., Ji, L., Gan, C., Lu, X., y Duan, H. (2014). “Discovery of clinical pathway patterns from event logs using probabilistic topic models”, *Journal of biomedical informatics*, vol. 4, pp. 39-57.

Kayi, E. S., Yadav, K., y Choi, H. A. (2013). “Topic modeling based classification of clinical reports”, *In 51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop,* pp. 67-73.

van Altena, A. J., Moerland, P. D., Zwinderman, A. H., y Olabarriaga, S. D. (2016). “Understanding big data themes from scientific biomedical literature through topic modeling”, *Journal of Big Data*, vol. 3, pp. 1-21.

D. M. Blei, A. Y. Ng y M. I. Jordan. (2003). “Latent dirichlet allocation”, *Journal of machine Learning research*, vol. 3, pp. 993–1022.

Qiang, J., Qian, Z., Li, Y., Yuan, Y., y Wu, X. (2020). “Short text topic modeling techniques, applications, and performance: a survey”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, pp. 1427-1445.

Quan, X., Kit, C., Ge, Y., y Pan, S. J. (2015). “Short and sparse text topic modeling via self-aggregation”, *In Twenty-fourth international joint conference on artificial intelligence*, pp. 2270-2276.

Fuentes-Pineda, G., y Meza-Ruiz, I. V. (2019). “Topic discovery in massive text corpora based on min-hashing”, *Expert Systems with Applications*, vol. 136, pp. 62-72.

Capítulo 5

Record linkage – Un análisis comparativo de las métricas de similitud

María Josefa Somodevilla García, Pierre Antoine Delice

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
Avenida San Claudio, 14 Sur, Ciudad Universitaria.
Puebla México.
{mariajsomodevilla, padelice}@gmail.com
https://lke.cs.buap.mx/doctorado/

Resumen. *Record Linkage* es uno de los métodos más utilizados para vincular registros. Una adecuada vinculación depende tanto de la calidad de los identificadores, las técnicas de indexación como de los métodos de comparación seleccionados. En la literatura, se han encontrado diversas métricas de similitud que abarcan desde la valoración de los errores de edición entre las cadenas de texto (Levenshtein, Jaro, Smith Waterman) hasta la composición de la estructura de las mismas (Coseno, N-Grama, Jaccard). En este trabajo, se experimentan las métricas de comparación en un conjunto de datos de personas que fueron sometidas a pruebas de COVID-19 con el fin de buscar duplicados. Usando un enfoque determinista del *Record Linkage*, el algoritmo de Jaro-Winkler arroja un mejor desempeño.

Palabras Clave: *Record Linkage*, algoritmo de comparación, medidas de similitud.

1 Introducción

Record Linkage tiene como objetivo vincular registros entre dos o más conjuntos de datos, basado en las entidades nombradas, cuando es aplicado a una base de datos se llama *Deduplication* o búsqueda de duplicados. Esta metodología remonta antes de la era de la computación moderna, donde científicos en el área de la salud pública buscaban explicar el origen social de ciertas enfermedades. Principalmente con los trabajos de William Farr (1803-1887) y Pedro Stocks (1944), la idea era combinar atributos de diferentes fuentes de información para entender la relación entre la estructura social y las enfermedades (Eyler, 1973; Lunde, 1975).

Retomado por Halbert Dunn (1946), quien usó por primera vez el concepto de *Record Linkage,* que a continuación llamaremos "vinculación de registros"; buscaba crear un libro de vida de las personas. La idea era documentar los eventos de vida de la población

empezando por las condiciones del nacimiento y concluyendo por las de mortalidad. Con esta integración, Dunn reveló que es capaz de generar información que sería imposible tener si se tuviese que producirla de manera tradicional. Desde entonces, se han visto distintas aplicaciones de esta metodología a lo largo de la historia, países como Alemania, Australia, Canadá, Estados Unidos entre otros tienen bien operacionalizada esta actividad en sus dependencias de estadísticas.

Estas dependencias han contribuido de manera significativa en mejorar la metodología de vinculación de registros, mediante sus numerosas publicaciones y colaboración con la academia. Sin embargo, las distintas comunidades científicas involucradas en el abordaje de este concepto lo han estudiado desde diferentes perspectivas teóricas, por lo que se han identificado dos grandes enfoques de vinculación. La primera, basada en métodos deterministas, consiste en una vinculación comparando con exactitud los pares de registros para establecer la existencia de correspondencia o no. Mientras que la otra, basada en métodos probabilistas, hace uso de un enfoque no supervisado para determinar el grado de similitud entre los pares de registros comparados (Cohen et al., 2003).

Existen un sinnúmero de investigaciones publicadas en las dos áreas, donde el objetivo consiste en buscar mejores estimadores de comparación entre las variables de identificación con el fin de maximizar la vinculación. Es decir, entre más alta la similitud arrojada por un algoritmo mayor es la probabilidad de vinculación. Para eso, la elección de las métricas de comparación tiene que ser analizada y estar en concordancia con las características de las variables a comparar.

Las métricas de comparación juegan un papel fundamental en la clasificación de los registros ya que ofrecen un análisis detallado respecto al grado de similitud entre estos, y a su vez determinan el dominio probable de la vinculación. Por lo general, son algoritmos basados en reglas, aplicados en su mayoría en estructuras de textos en inglés, así que el conocimiento adquirido por dichos algoritmos debe ser evaluado en otros contextos. Como es el caso del presente estudio, aplicados en conjuntos de datos con cadenas de textos del idioma español; el objetivo consiste en evaluar el desempeño de estas métricas y analizar las más adecuadas para la comparación y clasificación de registros a vincular.

El trabajo está organizado de la siguiente manera, primero se hará una breve descripción del método de vinculación de registros, para posteriormente analizar las métricas de similitud propuestas en la literatura. Finalmente, se concluye el análisis usando un conjunto de datos reales para comparar las métricas mencionadas.

2 Metodología de vinculación de registros

La metodología para la vinculación de registros está dividida en varias etapas. La primera inicia con el preprocesamiento y la extracción de las características para la *identificación* y *estandarización* del campo llave. En esta etapa, se busca definir las variables de identificación entre las bases de datos. Cuando no existe un identificador único, es importante considerar un conjunto de descriptores, donde cada variable tiene un peso específico en la identificación de la entidad (Ariel et al., 2014).

La *estandarización* consiste en homologar estas variables con el fin de reducir el costo computacional al momento de la comparación. Si bien no existe un manual que describa los pasos a seguir, pero se asume que se deben de reducir la cantidad de caracteres no deseados en las cadenas de textos, eliminar los espacios en blanco, así como las puntuaciones irrelevantes, entre otras. Posteriormente, se procede a la indexación que consiste en generar los pares de registros que serán analizados a detalle con las funciones de comparación. En este último paso, la vinculación se culmina clasificando los registros comparados en vinculado o no vinculado. Por el momento, nos detendremos en la clasificación binaria ya que pretendemos usar un enfoque determinista en este trabajo.

2.1 Enfoque determinista de vinculación

De acuerdo con el enfoque determinista, las variables seleccionadas para el proceso de vinculación tienen el mismo grado de importancia ya que estamos en presencia de identificadores o llave única (p.ej. id oficial, pasaporte). Cuando concuerdan significa que los registros corresponden a la misma persona o dicho de otra manera están vinculados. Sin embargo, si por error no existe concordancia entre los registros, el proceso resulta en una no-vinculación (Ariel et al., 2014).

Una forma de generalizar este enfoque consiste en suponer que existe acuerdo o desacuerdo en un conjunto de variables $k = 1, 2, \dots, K$, (ecuación 1).

$$y_{kij} = \begin{cases} 1, si\ los\ pares(i,j)\ concuerden\ con\ las\ variables\ k \\ 0, de\ otra\ manera \end{cases} \quad (1)$$

La comparación de todas las variables para para los registros (i, j) se escribe en la ecuación 2:

$$f_{ij} = \sum_{k} \gamma_{kij} \quad (2)$$

La regla de decisión para seleccionar o no un par de registros vinculados (i, j) se presenta en la ecuación 3.

$$x_{ij} = \begin{cases} 1, f_{ij} \geq \beta \\ 0, de\ otra\ manera \end{cases} \quad (3)$$

Donde $\beta \in \{k - n, \dots, k - 1, k\}$ y n representa la cantidad de variables en las que haya desacuerdos, $0 \leq n < k$. Este modelo considera que la vinculación entre un par de registros (i, j) ocurre si los valores concuerdan al menos en $k - n$ variables. Cuando el vínculo resulta con todas las variables k entonces $n = 0$.

3 Métricas basadas en distancia

Las métricas basadas en distancia permiten comparar dos cadenas de texto resultando en un valor numérico, mientras mayor sea el número más distantes son los conjuntos de textos. Esto constituye la base para entender las métricas de similitud, una vez normalizada, esta distancia se convierte en una función de similitud. Cada una de estas métricas resuelve un aspecto particular en la comparación. Algunos se basan en edición de caracteres, donde miden el número de operaciones necesarias para transformar una cadena de texto en otra. Mientras que otros buscan establecer semejanza en la estructura de los campos (token). Basado en ello, se presentan en la tabla 1 algunos de estos algoritmos: Levenshtein (1966), Damerau-Levenshtein, Jaro (1989), Jaro-Winkler (1990), Jaccard, Q-Grama, Cosine, Smith-Waterman (1981), *Longest Commons Substring* y *Longest Common Subsequence*.

Tabla 1. Métricas de comparación

Métricas	Representación	Complejidad	Consideraciones
Levenshtein (1966)	$d_l(A,B) = 1.0 - \frac{Levenshtein(A,B)}{max(\|A\|,\|B\|)}$	$O(\|A\| * \|B\|)$	Es el número de inserción, eliminación y sustitución necesarias para convertir una cadena de texto A en otra B.
Damerau-Levenshtein	$d_{dl}(A,B) \leq d_l(A,B)$	$O(\|A\| * \|B\|)$	Además del número de inserción, eliminación y sustitución, incluye la transposición entre caracteres adyacentes.
Jaro-Winkler	$d_{jw}(A,B) = d_j(A,B) + \frac{c}{10}\left(1 - d_j(A,B)\right)$	$O(\|A\| * \|B\|)$	Es una versión mejorada del algoritmo de Jaro, asignando mayor probabilidad a las cadenas de textos cuyas primeras letras coinciden
Q-Grama	$d_{qg}(A,B,q) = \frac{\|Qgram(A,q) \cap Qgram(B,q)\|}{max(\|Qgram(A,q), Qgram(B,q)\|)}$	$O(\|A\| + \|B\|)$	Conocido como n-grama divide las cadenas de texto entre subsecuencias de longitud q, luego se hace una comparación de los subconjuntos para determinar el grado de similitud
Coseno	$d_{cs}(A,B) = \frac{A \cdot B}{\|\|A\|\| \, \|\|B\|\|}$	$O(\|A\| + \|B\|)$	Prioriza el contenido entre las cadenas de texto en lugar del orden. La medición contempla el número de caracteres únicos y espacios entre las cadenas
Jaccard	$d_{jc}(A,B) = \frac{\|A \cap B\|}{\|A \cup B\|}$	$O(\|A\| + \|B\|)$	Es una medida basada en tokens y es definida como la relación de presencia-ausencia entre el número de caracteres comunes y el número total de caracteres en dos cadenas de texto
Longest Common Sustring (LCS)	$l_c = \sum_{i=1}^{n} \|s_{c_i}\|$ n es el número ro de particiones comunes.	$O(\|A\| * \|B\|)$	Consiste en encontrar las particiones (n-grama) comunes más largas de cadenas de texto entre S_a y S_b
Longest Common Subsequence (LCSubSeq)	$LCS(X_{i,}Y_{j,}) = \begin{cases} 0 \; si \; i = 0 \; o \; j = 0 \\ LCS(X_{i-1,}Y_{j-1,})^{x_i} \; si \; i,j > 0 \; y \; x_i = y_j \\ \max\{LCS(X_{i,}Y_{j-1,}), LCS(X_{i-1,}Y_{j,})\} \; si \; i,j > 0 \; y \end{cases}$	$O(\|A\| * \|B\|)$	Consiste en encontrar la mayor subsecuencia de caracteres comunes entre dos campos.
Smith-Waterson (1981)	$H_{ij} = \begin{cases} H_{i-1,j-1} + s(a_{i,}b_j), \\ max_{k \geq 1}\{H_{i-k,j} - W_k\} \\ max_{k \geq 1}\{H_{i,j-l} - W_i\} \end{cases}$	$O(\|A\| * \|B\|)$	A diferencia del algoritmo de Levenshtein, este asigna un puntaje a las diferencias y a los caracteres específicos. Esto permite un sistema de puntajes donde caracteres similares tendrán puntajes bajos

4 Experimentos

4.1 Datos

Para este experimento, se utiliza un conjunto de datos administrativos de personas que fueron sometidas a pruebas COVID-19 entre marzo 2020 a diciembre 2021, cuenta con 175,829 registros e incluye datos personales (nombre, apellidos, dirección, entre otros) que para efecto de privacidad no se presentarán en los resultados ni en los ejemplos presentados.

Los datos personales son utilizados como llave para la búsqueda de duplicidad, por lo que es una excelente base de datos para la aplicación de las métricas de comparación. En este caso, la presencia de registros duplicados no es un problema ya que por la dinámica de la enfermedad del COVID-19, se recomienda a la población repetir la prueba para descartar o confirmar la presencia del virus, por lo que se asume que un mismo registro ingresado con diferentes fechas cumple con este criterio.

Bajo esta perspectiva, conocer el número de duplicados es importante para evaluar el acceso a pruebas o el control de la enfermedad. Así mismo, regiones con más registros duplicados implica un mayor control del virus, si bien no es el objeto de este estudio sin embargo es parte del alcance que tiene esta metodología.

4.2 Preprocesamiento

Como se señaló anteriormente, la tarea principal del pre-procesamiento consiste en transformar datos heterogéneos en un conjunto estandarizado y consistente para posterior análisis. En este caso, definimos 6 atributos (nombre, apellido paterno, apellido materno, genero, edad, lugar de nacimiento) para definir la identidad de una persona, por lo que se procede a su homologación y estandarización.

Figura 1. Ejemplo de la estandarización de las variables de identificación

id	Nombre	Apellidos	Fec. Nac.	Dirección
R1	Juan_Antonio’	Reyes	1980/05/01	42 Av. Magallanes, No. 25 Cd. Mx 03458
R2	Juana Antonio	Reyes	01/05/1980	42 avenida Magallanes, int. 25 CdMx CP 03458

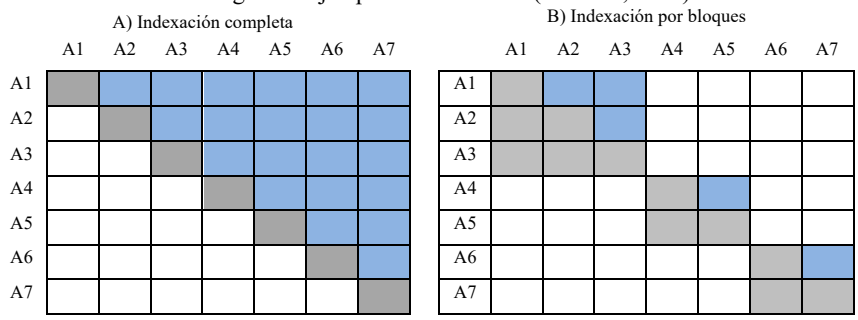
R1	Juan Antonio	Reyes	01/05/1980	Av. Magallanes #42, interior 25 Ciudad de México C.P.: 03458
R2	Juan Antonio	Reyes	01/05/1980	Av. Magallanes #42, interior 25 Ciudad de México C.P.: 03458

En la figura 1 se muestra un ejemplo del proceso de estandarización, como la eliminación de caracteres y espacios no esenciales, la homologación de la fecha de nacimiento y de la dirección.

4.3 Indexación

La indexación constituye un filtro para reducir el costo de operación del algoritmo, es el proceso por el cual, se retienen los pares de registros que son susceptibles de ser vinculados, agrupando los registros que suelen compartir los mismos descriptores. La indexación completa compara cada registro con los demás, es decir para una base de datos $D\ de\ 175{,}829$ registros, la cantidad de pares posibles seria $S = |D| * \frac{|D|-1}{2}$, lo que aproxima a un costo cuadrático de (15×10^9). Una alternativa es la indexación por bloqueo (*blocking*) que consiste en comparar los campos cuyas características se aproximan (ver figura 2).

Figura 2. Ejemplo de indexación (Christen, 2012)



Con el fin de incrementar el espacio de comparación, usamos varias combinaciones de bloqueo. Empezando por el nombre completo (nombre y apellidos) contamos con 23,827 posibles duplicados, mientras más características se agregan a esta combinación (fecnaci: fecha de nacimiento) disminuye el espacio de comparación a 18,133.

Considerando solo variables de identificación públicas como es la fecha de nacimiento, el municipio de residencia (mpioresi), la condición de pueblos autóctonos (esindige), la ocupación (ocupacio), así como la condición de habla de lenguas originarias (hableind), el número de registros a comparar aumentan a 46,851 (ver tabla 2).

Tabla 2. Indexación usando diferentes combinaciones de los identificadores

	Combinación de variables	Registros
A	nombre completo	23,827
B	nombre completo +fecnaci	18,200
C	nombre completo + fecnaci + sexo	18,133
D	fecnaci + mpioresi + esindige + ocupacio + hableind	46,851
	Total ($A \cup B \cup C \cup D$)	54,787

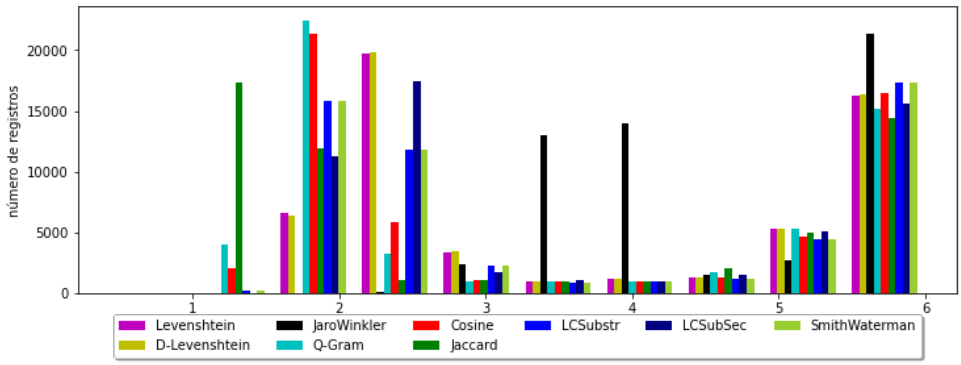
Finalmente, sumamos los diferentes bloques cuyo resultado arroja un total de 54,787 registros que serán analizados detalladamente por las métricas de comparación.

4.4 Comparación

Para la comparación, se seleccionan 6 variables de identificación: nombre, apellido paterno, apellido materno, fecha de nacimiento, dirección y la Clave Única de Registro de Población (CURP). Ahí, se agregó la CURP en vez del sexo ya que presenta más variabilidades. Estos descriptores se comparan entre sí y el resultado se suma arrojando un máximo de 6 puntos para los registros que tienen un empate perfecto, 0 cuando no existe similitud entre los registros.

Los resultados indican diferencia entre los algoritmos, como se puede observar en la figura 3, las métricas como Q-grama, Jaccard y Coseno reportan un número considerable de registros con poca o nula similitud, ya que tienen una distribución más cargada a la izquierda. Al basarse en la estructura de las cadenas de texto, son aptos para textos largos.

Figura 3. Comparación de las métricas de similitud



Algoritmos como Jaro-Winkler tienden a asignar una alta puntuación respecto a la comparación. Así mismo, por ser un enfoque determinista de vinculación, todas las variables tienen la misma importancia en la comparación, por lo que la métrica con el valor más alto nos aproxima a la detección de los duplicados (Ariel et al., 2014). Esto se observa en la tabla 3, al discretizar las métricas considerando combinaciones de identificadores mayor a 5 como duplicados, las que se encuentran entre 4 y 6 como probables duplicados finalmente las que suman menos que 4 como no duplicados. Se observa que los algoritmos de Jaccard, Q-grama y Smith Waterman arrojan menor cantidad de duplicados mientras que Jaro-Winkler capta la mayor cantidad de duplicados.

Tabla 3. Número de registros duplicados por métricas de similitud

Algoritmo	**Duplicados**	**Posibles duplicados**	**No duplicados**
Jaccard	18,630	3,537	32,620
Q-grama	19,621	3,198	31,968
Smith Waterman	20,198	2,550	32,039
Cosine	20,357	2,537	31,779
Levenshtein	21,033	2,567	31,187

Algoritmo	Duplicados	Posibles duplicados	No duplicados
Damerau-Levenshtein	21,050	2,578	31,159
LCSubs	21,280	2,271	31,236
LCSubSec	21,280	2,271	31,236
Jaro Winkler	23,487	6,535	24,765

5 Conclusiones

En este trabajo, se hizo una breve revisión de los principales métodos de comparación existentes en la literatura para evaluar el grado de similitud entre dos cadenas de texto. Como señalado, cada una de estas métricas pone énfasis en un aspecto particular de la comparación; se distinguen 2 clases de métricas: las basadas en edición de caracteres y las basadas en token. Estos juegan un papel fundamental en la vinculación de los registros ya que es el método por el cual se determina si 2 pares de registros son iguales o no. Así como mencionado, la vinculación plantea diversos retos a lo largo de su ejecución, por lo que implica analizar cada etapa, haciendo una revisión de las principales técnicas utilizadas en cada paso (Wang & Dong, 2020).

Usando un conjunto de datos de 175,829 personas que fueron sometidas a pruebas COVID-19 para buscar posibles duplicados, considerando variables de identificadores como nombre, apellidos, lugar de residencia, fecha de nacimiento y la CURP, se observa que la métrica de similitud de Jaro-Winkler muestra un mejor desempeño comparando con las demás métricas analizadas en esta investigación.

Referencias

Ariel, A., Bakker, B. F. M., Groot, M. de, Grootheest, G. van, Laan, J. van der, Smit, J. H., & Verkerk, B. (2014). Record linkage in health data: A simulation study. *Statistics Netherlands*, 64.

Christen, P. (2012). Indexing. En P. Christen (Ed.), *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection* (pp. 69–100). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-31164-2_4

Cohen, W. W., Ravikumar, P., & Fienberg, S. E. (2003). A Comparison of String Distance Metrics for Name-Matching Tasks. *IIWeb*.

Eyler, J. M. (1973). William Farr on the Cholera: The Sanitarian's Disease Theory and. The Statistician's Method. *Journal of the History of Medicine*, 79–100.

Jaccard, P. (1912). The distribution of the Flora in the Alpine zone 1. *New Phytologist*, *11*(2), 37–50. https://doi.org/10.1111/j.1469-8137.1912.tb05611.x

Jaro, M. A. (1989). Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association*, *84*(406), 414–420. JSTOR. https://doi.org/10.2307/2289924

Levenshtein, V. I. (1965). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*, *10*, 707–710.

Lunde, A. S. (1975). The Birth Number Concept and Record Linkage. *American Journal of Public Health*, *65*(11), 1165–1169.

Smith, T. F., & Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, *147*(1), 195–197. https://doi.org/10.1016/0022-2836(81)90087-5

Wang, J., & Dong, Y. (2020). Measurement of Text Similarity: A Survey. *Information*, *11*(9). https://doi.org/10.3390/info11090421

Winkler, W. E. (1990). *String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage*.

Winkler, W. E., & Thibaudeau, Y. (1987). An application of the Fellegi-Sunter model of record linkage to the 1990 US decennial census. En *Technical report, US Bureau of the Census*.

Capítulo 6

Interfaz web para recuperar información de Onto4UPPue, una ontología del repositorio institucional de la UPPue

Ana Laura Lezama Sánchez[1], María Auxilio Medina Nieto[2], Mireya Tovar Vidal[1]

[1] Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
[2] Universidad Politécnica de Puebla

ana.lezama@alumno.buap.mx, maria.medina@uppuebla.edu.mx,
mireya.tovar@correo.buap.mx

Resumen. Este documento propone el uso de una ontología de dominio restringido construida de forma manual, cuyas instancias agrupan información de tesis del Departamento de Posgrado de la Universidad Politécnica de Puebla. La ontología es una adaptación de *Onto4AIR*, la cual modela el conocimiento de dominio y operativo de los repositorios institucionales de acuerdo con los lineamientos generales y técnicos del Consejo Nacional de Ciencia y Tecnología. Se presenta una interfaz accesible vía web, por medio de la cual se logró realizar consultas, los resultados preliminares de usabilidad reportan que la interfaz es útil para usuarios expertos y no expertos en tecnologías semánticas. Con la implementación de la interfaz se aporta una herramienta capaz de dar al usuario información consistente a sus necesidades, dado que hace uso de herramientas semánticas.

Palabras Clave: datos abiertos, datos abiertos enlazados, ontologías, *SPARQL*, pruebas de usabilidad.

1 Introducción

Un Repositorio Institucional (*RI*) es una plataforma digital centralizada que almacena la producción científica y académica de una institución educativa. Soporta tareas como búsqueda, almacenamiento, distribución, difusión y recuperación de información (Bustos-González et al., 2007); contiene mecanismos para importar, identificar, almacenar, preservar y exportar un conjunto de documentos digitales descritos mediante etiquetas o metadatos que facilitan su recuperación. Las colecciones o conjunto de documentos incluyen tanto la producción científica como: artículos, tesis, objetos de aprendizaje, así como documentos administrativos que generan una institución, en formatos diferentes como textos, presentaciones o registros audiovisuales. En México, los *RIs* se centran en la producción científica que se distribuye bajo los términos de políticas de acceso abierto, se podrían considerar como fuentes de datos válidos para obtener indicadores. En un *RI* pueden surgir problemas en la administración de los datos como 1) la inconsistencia, que radica en

el incumplimiento de reglas y/o restricciones establecidas previamente, y 2) datos incompletos o no validados (Pérez López, 2004). Las ontologías son una alternativa para afrontar estos problemas.

Una ontología se define como una especificación explícita y formal de una conceptualización compartida (Gruber, 1995), este tipo de recurso semántico está formado por conceptos o clases, relaciones, instancias, atributos, axiomas, restricciones, reglas y eventos. Las ontologías de dominio se consideran un sistema de representación del conocimiento que organiza conceptos de algún área o dominio específico en estructuras taxonómicas y no taxonómicas (Tovar Vidal et al., 2015). Una de las tareas que se realizan con las ontologías es el poblado de estas. Esta tarea consiste en agregar instancias a un modelo ontológico. Requiere tener previamente definido un modelo semántico, de manera que sea necesario modelar el dominio de conocimiento y representarlo en alguno de los lenguajes existentes para este fin (*World Wide Web Consortium*) y luego poblarlo con la información obtenida con anterioridad (Abello Diaz, 2015). El término metadato, según (Senso, 2003), fue acuñado por Jack Myers en la década de los 60 para describir conjuntos de datos. La primera aceptación que se le dio (y actualmente la más extendida) fue la de datos sobre el dato, ya que proporciona la información mínima necesaria para identificar un recurso. Un metadato puede incluir información descriptiva sobre el contexto, calidad y condición o características del dato.

En este documento se presenta una ontología que modela conocimiento operativo y de dominio del Repositorio Institucional de la Universidad Politécnica de Puebla (*RI-UPPue*) denominada *Onto4UPPue*. La cual se pobló de forma manual usando el programa de edición de ontologías *protégé* (Musen, 2015), con instancias que corresponden a tesis de posgrado. Se presenta una interfaz de consulta diseñada para usuarios expertos y no expertos en tecnologías semánticas. El objetivo de este trabajo es el desarrollo de una interfaz de búsqueda semántica que forme parte del repositorio institucional de la *UPPue*.

El documento está estructurado como sigue: La sección 2 contiene los trabajos relacionados, la sección 3 describe la interfaz propuesta, la sección 4 presenta los resultados preliminares de usabilidad y en la sección 5 se discuten las conclusiones el trabajo a futuro y finalmente las referencias.

2 Estado del arte

Esta sección describe los trabajos relacionados con temas afines al diseño y desarrollo de la interfaz descrita en este documento. Se presentan artículos que hacen referencia al acceso abierto.

En (Mazzanti et al., 2018) exponen los resultados preliminares obtenidos, enlazar publicaciones científicas de *DSpace* con *datasets* del Sistema Nacional de Datos Biológicos (*SNDB*) con datos primarios citados en ellas utilizando *Resource Description Framework (RDF)* y de esta manera resolver la interoperabilidad a nivel semántico de vocabularios

entre repositorios documentales. Los autores utilizaron tecnologías de la Web Semántica, estándares recomendados por *W3C* y se basaron en el enfoque propuesto por datos abiertos enlazados (*LOD* por sus siglas en inglés). El desarrollo realizado generó la publicación de datos en *RDF* accesibles a través de *SPARQL endpoint*. Los autores definieron una consulta integrada *SPARQL* que involucra dos o más conjuntos de datos *RDF* y resolvió la vinculación de publicaciones científicas con datos primarios.

En (Fermoso et al., 2019) el objetivo es promover la transformación en *Linked Open Data* de los catálogos bibliográficos tradicionales y por lo tanto dar a conocer la información sobre autoridades y figuras ilustres que se almacenan en un catálogo bibliográfico de la biblioteca de la Universidad Pontificia de Salamanca. Para ello, los autores aplicaron formatos especializados de datos abiertos enlazados (*LOD*) para el ámbito bibliográfico. Los autores utilizaron el formato *BIBFRAME* que está basado en la catalogación basado en *LOD* especialmente diseñado para registros bibliográficos y archivísticos. *BIBFRAME* pretende diferenciar los conceptos como *work* e *instance*.

En (Duran et al., 2020) exponen un sistema de recomendación de recursos educativos, que se basó en el desempeño del cumplimiento de metas de aprendizaje y aprovecha el conocimiento de los datos abiertos enlazados. La metodología propuesta incluyó cinco fases que consistieron en definición de una arquitectura y lógica del sistema, determinación del modelo de conocimiento, definición de un modelo de usuario, determinación de un mecanismo de filtrado, el cual señala los aspectos computacionales del sistema para generar las recomendaciones y experimentación, a través de la cual se mide la pertinencia de las recomendaciones.

En (Torre-Bastida et al., 2015) realizan un análisis de las principales iniciativas relacionadas con datos enlazados referidas a bibliotecas. Los autores destacan el potencial de la aplicación de estas técnicas en el área bibliotecaria ya que pueden resolver gran parte de los retos asociados a la gestión de datos bibliotecarios. Además de incorporar la web semántica y *linked open data cloud* ya que pueden ayudar a las bibliotecas a fomentar la publicación, interconexión y compartición de datos.

En (Samec et al., 2020) el trabajo de los autores tiene por objetivo hacer accesible y abiertos los datos, la comunidad científica, de una base de datos llamada *Southwest Atlantic Benthic Invertebrates* que almacena los datos de invertebrados de la región y publicaciones taxonómicas por medio de Datos Abiertos Enlazados y de esta manera fuera interoperable con bases de datos de referencia global desarrollando micro-servicios *SPARQL*.

3 Interfaz de consulta para onto4UPPue

En este documento se propone una interfaz accesible vía web que permite consultar la ontología *Onto4UPPue*, la cual contiene datos de tesis de posgrado de la Universidad Politécnica de Puebla (*UPPue*). *Onto4UPPue* se deriva de la ontología *Onto4AIR*, la cual forma parte de la estrategia denominada *Linked Open Data for All Institutional Repositories*

(LOD4AIR), constituye un vocabulario controlado, una representación de conocimiento formal, no ambigua que puede compartirse y reutilizarse entre usuarios y computadoras.
Onto4AIR modela conocimiento de dominio y operativo relacionado con los usuarios, las políticas de distribución de *AA* de los contenidos y la funcionalidad de los *RIs* de conformidad con los lineamientos generales y técnicos del Consejo Nacional de Ciencia y Tecnología (*CONACYT*) Convocatoria I0028-2016-04.
Los conceptos y relaciones en *Onto4UPPue* tratan temáticas como tipos de usuario, organización de contenidos, políticas de distribución de acceso abierto, reglas de operación, procedimientos de interoperabilidad y mantenimiento (Medina et al., 2017). A manera de ejemplo, la Figura 1 muestra parte de la taxonomía de clases de Onto4UPPue, es de interés mostrar aquellas que se relacionan con las clases Archivo y Tesis, dado que las instancias de *Onto4UPPue* pertenecen a estas clases.
Onto4UPPue se caracteriza por integrar elementos del formato de metadatos estándar *Dublic Core* como propiedades de datos de la clase Archivo. El propósito es validar automáticamente la consistencia de los datos y extender los mecanismos de consulta con tecnologías semánticas.

Figura 1 Taxonomía de la clase Archivo de la ontología Onto4UPPue

La interfaz de consulta para *Onto4UPPue* se diseñó con la finalidad de que cualquier usuario pueda consultar su información sin importar si cuenta con conocimientos o no de tecnologías semánticas, está implementada en el lenguaje *PHP* y utiliza el *SPARQL endpoint* denominado *arc2*, el cual provee la comunicación entre la interfaz y la ontología. El desarrollo de la interfaz fue hecho bajo la técnica de ingeniería de software en cascada, donde el desarrollo del software se concibe como un conjunto de etapas que se ejecutan una tras otra. En este trabajo se siguieron las etapas de requisitos, diseño, implementación y evaluación. La ontología se pobló con tesis de posgrados de diferentes instituciones de la república mexicana como BUAP, UDLAP, UNAM, UPP entre otras con un total de 99 tesis de posgrados almacenadas en la ontología. Las consultas que el usuario realiza se consideran de dos tipos: 1) predefinidas o basadas en formularios como en el caso de búsqueda por autor (2).

La Figura 2 muestra el formulario para buscar tesis por autor, el único campo obligatorio es el apellido paterno.

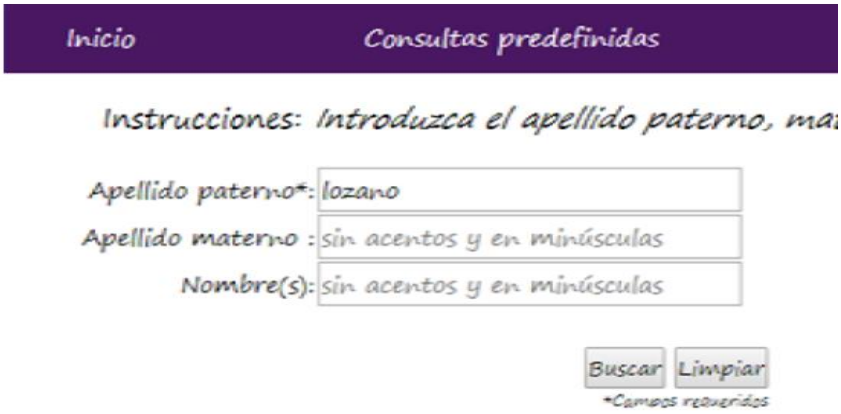


Figura 2 Formulario para buscar tesis por autor

La Figura 3 muestra los metadatos y resultados asociados a la consulta búsqueda por autor; al dar clic en el enlace "Ubicación" se accede al texto completo de la tesis. En el caso de que la consulta no recupere datos, la interfaz muestra un mensaje.

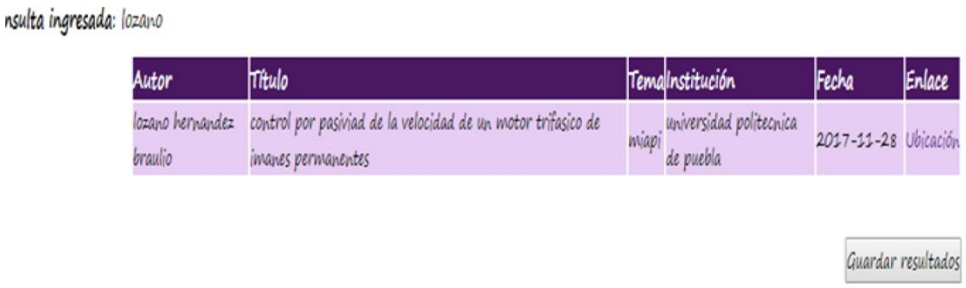


Figura 3 Resultado de una búsqueda por autor

La Figura 4 detalla el formulario de búsqueda por fechas, existen además las búsquedas por título, asesor, tema y colaboradores y 2) libres cuando el usuario introduce una consulta en *SPARQL* directamente (ver Figura 5).

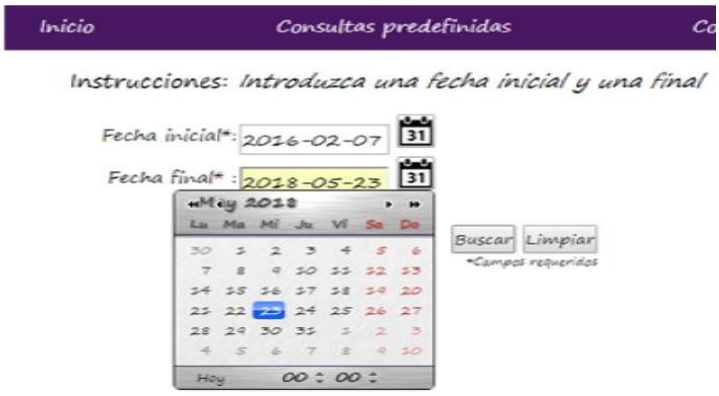


Figura 4 Formulario para la búsqueda de tesis por fechas

En los dos tipos de consultas se implementó un botón para guardar los datos de búsqueda y los resultados, al hacer clic se descarga un documento de texto plano como muestra la Figura 6. La Figura 7 se expone el contenido del archivo de texto denominado "consultaRealizada.txt". Los resultados cuando se realiza una búsqueda por fechas (ver Figura 4), se muestran en la Figura 8. Los resultados de la consulta que se realiza en la Figura 5, se muestran en la Figura 9, donde se recuperan las tesis de los autores cuyo nombre completo contenga la cadena "Martínez", el título incluya la palabra "algoritmo", con la participación de un colaborador y la fecha de presentación de la tesis sea anterior al 31 de diciembre del 2015. Notar que en la ontología no se almacenan caracteres especiales como acentos, propios del español. Para consultar cualquier ontología en *SPARQL* se utiliza un espacio de nombres que permite encontrar la ubicación en internet. Con la finalidad de reutilizar la ubicación de la ontología en la consulta, se utiliza un prefijo *(PREFIX)* que permite su sustitución, en cualquier sección de la consulta. En este caso se utiliza el prefijo ni y su valor es: *http://www.semanticweb.org/ontologies/2017/0/Ontology1484677652201.owl#*.

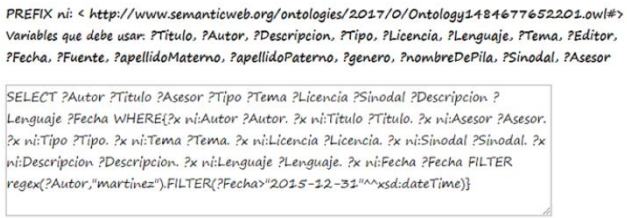


Figura 5 Campo de texto para insertar la primera consulta en *SPARQL*

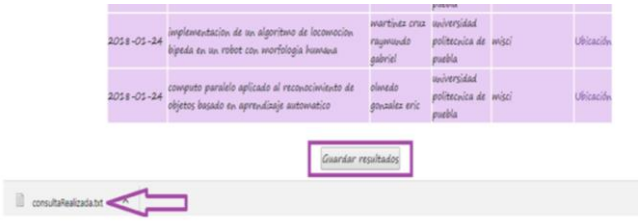


Figura 6. Icono para descargar el archivo de texto con los resultados

Figura 7. Contenido del documento de texto consultaRealizada.txt

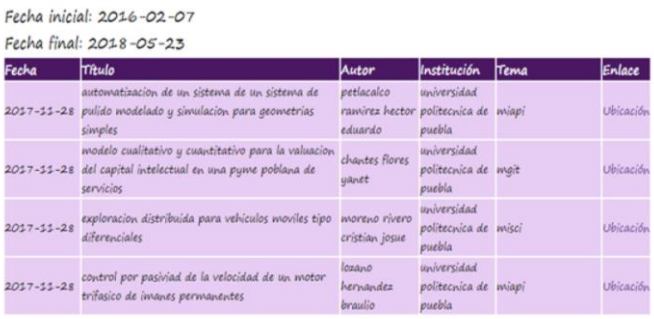


Figura 8. Resultado de una búsqueda por fechas.

Autor	martinez cruz raymundo gabriel
Título	implementacion de un algoritmo de locomocion bipeda en un robot con morfologia humana
Asesor	benitez ruiz antonio
Tipo	tesis de maestria
Tema	misci
Licencia	licencia 25 creative commons cc bc nd nd 25 mx
Sinodal	de la calleja mora jorge medina nieto maria auxilio
Descripción	realizar pruebas en la plataforma bioloid premium kit y documentarlas
Lenguaje	espanol
Fecha	2018-01-24
Enlace	Ubicación

Figura 9. Resultado de la primera consulta en *SPARQL*

La Figura 10 muestra otro ejemplo de consulta en SPARQL, en este caso se buscan que un colaborador tenga el nombre o apellido “benitez” el título de la tesis incluya la palabra “robot” y que la fecha sea posterior a 31-12-2017. Los resultados de la consulta se muestran en la Figura 11. La Figura 12 muestra un tercer ejemplo de consulta libre, en este caso, se busca que el tema de la o las tesis sea “misci”. La Figura 13 muestra algunos de los resultados obtenidos.

PREFIX ni: < http://www.semanticweb.org/ontologies/2017/0/Ontology1484677652201.owl#>
Variables que debe usar: ?Titulo, ?Autor, ?Descripcion, ?Tipo, ?Licencia, ?Lenguaje, ?Tema, ?Editor, ?Fecha, ?Fuente, ?apellidoMaterno, ?apellidoPaterno, ?genero, ?nombreDePila, ?Sinodal, ?Asesor

```
SELECT ?Autor ?Titulo ?Asesor ?Tipo ?Tema ?Licencia ?Sinodal ?Descripcion ?
Lenguaje ?Fecha WHERE{?x ni:Autor ?Autor. ?x ni:Titulo ?Titulo. ?x ni:Asesor ?
Asesor. ?x ni:Tipo ?Tipo. ?x ni:Tema ?Tema. ?x ni:Licencia ?Licencia. ?x ni:Sinodal ?
Sinodal. ?x ni:Descripcion ?Descripcion. ?x ni:Lenguaje ?Lenguaje. ?x ni:Fecha ?Fecha
FILTER regex(?Asesor,"benitez").FILTER regex(?Titulo,"robot").FILTER(?
Fecha>"2017-12-31"^^xsd:dateTime)}
```

Buscar Limpiar
*Campos requeridos

Figura 10. Campo de texto con la segunda consulta en *SPARQL*

Autor	martinez cruz raymundo gabriel
Título	implementacion de un algoritmo de locomocion bipeda en un robot con morfologia humana
Asesor	benitez ruiz antonio
Tipo	tesis de maestria
Tema	misci
Licencia	licencia 25 creative commons cc bc nd nd 25 mx
Sinodal	de la calleja mora jorge medina nieto maria auxilio
Descripción	realizar pruebas en la plataforma bioloid premium kit y documentarlas
Lenguaje	espanol
Fecha	2018-01-24
Enlace	Ubicación

Figura 11. Resultado de la segunda consulta en *SPARQL*

PREFIX ni: < http://www.semanticweb.org/ontologies/2017/0/Ontology1484677652201.owl#>
Variables que debe usar: ?Titulo, ?Autor, ?Descripcion, ?Tipo, ?Licencia, ?Lenguaje, ?Tema, ?Editor, ?Fecha, ?Fuente, ?apellidoMaterno, ?apellidoPaterno, ?genero, ?nombreDePila, ?Sinodal, ?Asesor

```
SELECT ?Autor ?Titulo ?Asesor ?Tipo ?Tema ?Licencia ?Sinodal ?Descripcion ?
Lenguaje ?Fecha WHERE{?x ni:Autor ?Autor. ?x ni:Titulo ?Titulo. ?x ni:Asesor ?Asesor.
?x ni:Tipo ?Tipo. ?x ni:Tema ?Tema. ?x ni:Licencia ?Licencia. ?x ni:Sinodal ?Sinodal. ?x
ni:Descripcion ?Descripcion. ?x ni:Lenguaje ?Lenguaje. ?x ni:Fecha ?Fecha FILTER
regex(?Tema,"misci")}
```

Figura 12. Campo de texto con la segunda consulta en *SPARQL*

Autor	huitzil velasco ignacio
Título	sistema distribuido de videovigilancia para dispositivos moviles
Asesor	lopez dominguez eduardo
Tipo	tesis de maestria
Tema	misci
Licencia	licencia 25 creative commons cc bc nd nd 25 mx
Sinodal	de la calleja mora jorge medina nieto maria auxilio
Descripción	los sistemas de video vigilancia con base en el tipo de tecnicas para deteccion de objetos y tecnologia implementada en funcion de la escalabilidad y la comunicación se dividen en tres generaciones
Lenguaje	espanol
Fecha	2018-01-24
Enlace	Ubicación

Figura 13. Resultado de la tercera consulta en *SPARQL*

4 Resultados experimentales

Con el objetivo de medir la percepción de la usabilidad de los usuarios, se aplicó el cuestionario de satisfacción de usabilidad de *IBM*, compuesto por 19 preguntas (Senso et al., 2003), (Lewis, 1995). En el proceso de percepción de usabilidad se aplicó el cuestionario de satisfacción de usabilidad de *IBM* a 10 usuarios que llevaron a cabo la evaluación de la interfaz; cuatro del sexo femenino y seis del sexo masculino, cuatro estudiantes de maestría con conocimientos en ontologías, *SPARQL*, conocimiento de los *RIs*, medatados y aplicaciones web y seis estudiantes de licenciatura con conocimiento principalmente en aplicaciones web. Para tener conocimiento del perfil de cada usuario, se les solicitó información como fecha, hora, nombre, género, profesión, área, grado de estudios, además de responder a los siguientes cuestionamientos:

1. ¿Conoce *SPARQL*, su propósito y sintaxis?
2. ¿Sabe que es una ontología?

3. ¿Tiene conocimiento acerca de lo que es un repositorio institucional?
4. ¿Conoce que es un metadato?
5. ¿Puede definir correctamente que es una aplicación web?

El estudio se realizó en el Laboratorio de Experiencia de Usuario de la Universidad Politécnica de Puebla. Los instrumentos utilizados para llevar a cabo la evaluación fueron *"IBM usability satisfaction questionnaires"* (Senso et al., 2003), (Lewis, 1995) empleada para medir la percepción de una aplicación. Las preguntas que forman parte del cuestionario se presentan en la Tabla 1.

Tabla 1. Lista de preguntas para la evaluación de la usabilidad de la interfaz

	Cuestionario
1	En general, estoy satisfecho con lo fácil que es utilizar la interfaz
2	Es sencillo de utilizar la interfaz
3	Puedo efectivamente completar mi trabajo con la interfaz
4	Puedo terminar mi trabajo de forma rápida usando la interfaz
5	Soy capaz de completar de manera eficiente mi trabajo con la interfaz
6	Me siento cómodo con la interfaz
7	Es fácil aprender a utilizar esta interfaz
8	Creo que me convertí productivo rápidamente con la interfaz
9	La interfaz da mensajes de error que claramente me digan cómo solucionar los problemas
10	Cada vez que cometo un error al utilizar la interfaz, se puede recuperar fácil y rápidamente
11	La información proporcionada con la interfaz es clara
12	Es fácil encontrar la información que necesito
13	La información proporcionada por la interfaz es fácil de entender
14	La información es eficaz para ayudar a completar las tareas
15	La organización de la información en las pantallas de la interfaz está clara
16	La interfaz es agradable
17	Me gusta usar la interfaz
18	Esta interfaz tiene todas las funciones y capacidades que espero que tenga
19	En general, estoy satisfecho con la interfaz

Para dar un puntaje a cada una las preguntas de la Tabla 1, el usuario debe indicar su respuesta a través de una escala Likert de 7 puntos (Manuel et al., 2010). Los valores asociados a cada uno de estos puntos son:

1. Totalmente en desacuerdo
2. Moderadamente en desacuerdo
3. Un poco en desacuerdo
4. Neutral
5. Un poco de acuerdo
6. Moderadamente de acuerdo
7. Totalmente de acuerdo

Las preguntas de la Tabla 1 sirven de herramienta para verificar satisfacción general de la interfaz; para ello se calcula el promedio de las preguntas de la Tabla 1. El valor asignado por cada usuario a cada pregunta se muestra en la Figura 14, las filas reflejan el número de pregunta y las columnas el número de usuario. A cada usuario se le asignaron 3 tareas.

Primero se les indicó que realizaran una búsqueda por autor, asesor y sinodal, el usuario fue libre de insertar los apellidos que quisiera, para posteriormente obtener el resultado, a la vez de que lograron obtener el documento de texto que resguarda la consulta ingresada y los datos recuperados. Segundo se les pidió que insertaran fechas iniciales y finales para la búsqueda por rango de fechas, y posteriormente obtener el documento de texto que resguardará la fecha final, y los resultados asociados, así como también búsqueda por tema y título. Tercero para la búsqueda libre, se les solicitó que insertaran una consulta con sintaxis en *SPARQL*. Una vez completada la consulta se les pidió que oprimieran el botón buscar y obtuvieron los resultados de la consulta que cada usuario insertó, para que se percataran de que los datos recuperados pertenecen a la consulta insertada. En el caso de los usuarios con conocimientos en *SPARQL*, el tiempo que tomaron para evaluar la interfaz fue de 4 minutos máximo, y después de realizar las tareas recomendadas, proporcionaron sus opiniones con respecto al diseño, mientras que para el resto de los usuarios el tiempo utilizado fue mayor a 10 minutos, ya que al mismo tiempo de realizar las tareas requeridas proporcionaron sus comentarios. Una vez que los usuarios respondieron al cuestionario que se les proporcionó, se calcula la confiabilidad de este, a través del *Alfa Cronbach* (Senso et al., 2003). Esta medida se presenta en la ecuación 2, si la respuesta es próxima a 1, significa que el valor obtenido es consistente. En la Figura 14 se muestran las respuestas de los diez usuarios.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	SUMA
U1	4	3	4	4	4	3	3	3	6	3	3	3	6	4	3	4	4	3	3	70
U2	7	6	6	6	6	6	5	4	4	7	7	7	4	4	4	4	4	7	5	103
U3	7	6	4	3	4	5	3	5	6	6	4	4	7	5	6	4	4	4	5	92
U4	7	6	6	6	7	7	7	7	6	4	7	4	3	4	5	6	5	5	4	106
U5	7	7	6	4	7	7	6	7	7	4	7	5	5	5	5	5	7	7	7	115
U6	7	7	6	7	4	7	6	4	7	4	4	6	6	5	5	4	4	4	4	101
U7	7	7	7	7	4	7	7	7	7	7	7	7	7	7	7	5	6	4	4	121
U8	4	4	7	6	6	7	7	7	6	6	6	7	7	7	7	4	6	5	4	113
U9	6	4	7	7	6	7	4	7	7	5	4	7	4	4	5	6	4	3	4	101
U10	4	4	3	4	4	4	4	4	4	4	3	6	6	6	3	5	4	3	3	78

Figura 14 Respuestas correspondientes a los diez usuarios encuestados

$$\alpha = \frac{K}{K-1}\left[1 - \frac{\sum_{I=1}^{K} S_I^2}{S_T^2}\right] \qquad (1)$$

Donde:

- α *Alfa de Cronbach*
- *K* número de preguntas seleccionadas
- $\sum S_I^2$ Suma de la varianza de cada pregunta
- S_T^2 Varianza de la suma de las preguntas

Utilizando los datos de la Figura 14 y la ecuación 1, el resultado de la medida de consistencia de *Alfa de Cronbach* es:

$$\alpha = \frac{19}{18}[1 - \frac{37{,}62222}{258{,}8889}] = 0{,}90216 \qquad (2)$$

El valor obtenido al aplicar el *Alfa de Cronbach* (Ledesma et al., 2002) es cercano a 1 por lo tanto se considera que el nivel de confiabilidad es aceptable.

5 Conclusiones

En este artículo se presenta el desarrollo de una interfaz de consulta en un RI, capaz recuperar información que el usuario solicita de manera clara y con la posibilidad de tener acceso a ella. Se presentan dos tipos diferentes de consultas, la primera predefinidas que son las que el usuario sin conocimiento en tecnologías semánticas podrá usar, que son por autor, fecha, sinodal y asesor, y las libres, para usuarios expertos que podrán hacer cualquier tipo de consulta con sintaxis en *SPARQL*. Una vez construida la interfaz de consulta, se llevó a cabo la evaluación, usando *IBM usability satisfaction questionaires* para medir la usabilidad de la interfaz. Para ello se contó con la evaluación de 10 usuarios, 4 expertos en el área de *SPARQL*, ontologías y aplicaciones web y 6 sólo en aplicaciones web. Hasta ahora hemos demostrado que el uso de una ontología como medio para resguardar información del material académico producido dentro de nuestra institución otorga muy buenos resultados, ya que reduce la inconsistencia en los datos. Para el caso de usuarios con conocimiento en tecnologías semánticas, podrán obtener un mayor número de resultados usando sintaxis en *SPARQL*, asociados a sus consultas, ya que dentro de las consultas libres se pueden obtener metadatos que con las consultas predefinidas no es posible recuperar. Como trabajo a futuro se propone que los resultados asociados a cada consultan arrojen un resumen de la información que está mostrando, esto con la finalidad de que el usuario pueda valorar si la información es de utilidad, además de incorporar las funciones de agregar, eliminar y editar instancias a la ontología de forma automática, para aumentar las funciones de la interfaz de consulta.

Referencias

Bustos-González, A., & Fernández-Porcel, A. (2007). Directrices para la creación de repositorios institucionales en universidades y organizaciones de educación superior.

Pérez López, C. (2004). Técnicas de análisis multivariante de datos. Aplicaciones con SPSS, Madrid, Universidad Complutense de Madrid, 121-154.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing?. International journal of human-computer studies, 43(5-6), 907-928.

Tovar Vidal, M., Pinto Avendaño, D. E., Montes Rendón, A., González Serna, J. G., & Vilariño Ayala, D. (2015). Evaluation of ontological relations in corpora of restricted domain. Computación y Sistemas, 19(1), 135-149.

Abello Diaz, J. A. (2015). Obtener un método para la extracción de información a partir de documentos semiestructurados producidos al interior del Servicio Nacional de Aprendizaje SENA, permitiendo su publicación, reutilización e intercambio a través de la web semántica. Ingeniería de Sistemas.

Senso, J. A., & Rosa Piñero, A. D. L. (2003). The metadata concept: something more than description of electronic resources. Ciência da Informação, 32, 95-106.
Musen, M. A. (2015). The protégé project: a look back and a look forward. AI matters, 1(4), 4-12.
Medina, M. A., Sánchez, J. A., Cervantes, O., Medina, R. R. C., De la Calleja, M. J., & Benitez, A. (2017). Representación semántica de conocimiento operativo y de dominio para repositorios institucionales. Registro público del derecho de autor, México, (03-2017), 042511235500-01.
Senso, J. A., & Rosa Piñero, A. D. L. (2003). The metadata concept: something more than description of electronic resources. Ciência da Informação, 32, 95-106.
Lewis, J. R. (1995). IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. International Journal of Human-Computer Interaction, 7(1), 57-78.
Manuel, C., Meza, P., Lagues, K., & Yañez, S. (2010). Adaptación y validación de la Escala de Orientación a la Dominancia Social (SDO) en una muestra chilena. Universitas Psychologica, 9(1), 161-168.
Ledesma, R., Molina, G., & Valero, P. (2002). Análisis de consistencia interna mediante Alfa de Cronbach: un programa basado en gráficos dinámicos. Psico-USF, 7(2), 143-152.
Samec, G., Diez, M. E., Zàrate, M., Buckle, C., Lima, J., Jaramillo, R., ... & Mazzanti, R. (2020). Herramientas informáticas para el estudio de la biodiversidad utilizando datos abiertos enlazados. In XXII Workshop de Investigadores en Ciencias de la Computación (WICC 2020, El Calafate, Santa Cruz).
Torre-Bastida, A. I., González-Rodríguez, M., & Villar-Rodríguez, E. (2015). Datos abiertos enlazados (LOD) y su implantación en bibliotecas: iniciativas y tecnologías. Profesional de la información, 24(2), 113-120.
Duran, D. F., Chanchí, G. E., & Arciniegas, J. L. (2020). Sistema de recomendación de recursos educativos basado en metas de aprendizaje y razonamiento en datos abiertos enlazados. Revista Ibérica de Sistemas e Tecnologias de Informação, (E32), 1-13.
Fermoso, A. M., Manzano, M. I., Armero, A., & Hernández, A. (2019). Apertura y publicación de datos bibliográficos con formatos de datos abiertos. Aplicación a un caso práctico.
Mazzanti, R., Buckle, C., Zárate, M., & Samec, G. (2018) Integración de repositorios semánticos: un camino hacia los datos abiertos enlazados.

Capítulo 7

Estudio Comparativo de Métricas en Grafos e Hipergrafos para el estudio de problemas intratables

Yolanda Moyao Martínez, Luis Carlos Altamirano Robles, Darnes Vilariño Ayala

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

{yolanda.moyao, luis.altamirano, darnes.vilarino}@correo.buap.mx

Resumen. En el presente trabajo se presenta una revisión del estado actual de los trabajos relacionados con diferentes métricas para grafos y que pueden ser extendidas a hipergrafos, tales como ancho arbóreo, ancho de corte, ancho de ruta y ancho de clique. La motivación de este trabajo se debe al estudio de Gramáticas de Reemplazo de Hiperaristas (HRG), de tal forma que de manera conjunta la métrica y los parámetros adecuados puedan conducir a una solución de parámetro fijo tratable (FTP) para el análisis de las HRG.

Palabras Clave: Gramáticas de Reemplazo de Hiperaristas, Métricas Arbóreas y Parámetro Fijo Tratable, Problemas Intratables.

1 Introducción

En este trabajo de investigación se realiza una revisión de los trabajos relacionados a las diferentes métricas de grafos e hipergrafos, con el propósito de que, en trabajos a futuro, se identifiquen la métrica y parámetro adecuado para encontrar una solución Parámetro Fijo Tratable (FPT), por sus siglas en inglés, para el análisis de Gramáticas de Reemplazo de Hiperaristas (HRG), por sus siglas en inglés.

Hay problemas para los cuales la estructura de alguna instancia se describe mejor con hypergrafos que con grafos. Recientemente, diferentes algoritmos de aprendizaje de hipergrafos han demostrado ser eficaces en diferentes aplicaciones, tales como multimedia, bioinformática, recuperación de texto, entre otras. Esta eficacia es debido a la naturaleza de los hipergrafos, en el sentido de que una hiperarista conecta a varios vértices con lo que se pueden modelar diferentes relaciones de alto orden (Huang et al, 2015).

Diferentes problemas dentro de áreas como, bases de datos y satisfacción de restricciones pueden ser tratables, cuando cierta clase de hipergrafos asociados con instancias del problema, tienen un ancho hipertarbóreo acotado (Gottlob et al, 2002). Además, desde que se introducen los conceptos de descomposición hiperarbórea y ancho hiperarbóreo, se han aplicado para dar solución a diferentes problemas, tales como, Conteo de Soluciones

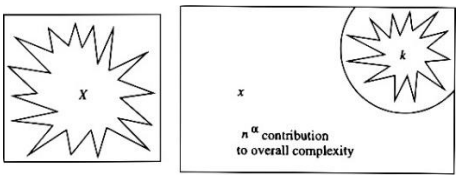
Estimadas, enumeración, optimización de restricciones y subastas combinatorias (Gottlob et al, 2014).

Las HRG, se han desarrollado como una extensión del concepto formal de Gramáticas Libres de Contexto. En éstas, el concepto de reescritura de símbolos por cadenas se generaliza al reemplazo de hiperaristas por hipergrafos, a pesar de que, en el caso de cadenas, el reemplazo de una hiperarista por otra suele ser un proceso sencillo. Para el caso de hipergrafos, es necesario encontrar un hipergrafo para ser reemplazado por otro hipergrafo (Lautemann, 1990).

Las HRG, las cuales son un formalismo para la generación de lenguajes de hipergrafos, el interés de este tipo de lenguaje es debido a las aplicaciones en diferentes áreas, tales como lingüística computacional, procesamiento del lenguaje natural, en particular en la comprensión y generación de lenguaje, en la traducción automática basada en semántica; así también en áreas como teoría de juegos, bases de datos, inteligencia artificial y diseño VLSI, entre otras (Peng et al, 2015, Peuser, 2018, Drews et al, 1997, Gallo et al, 1998, Moyano et al, 2016, y Hamm, 2019).

En varias disciplinas tales como Ciencias de la Computación, Ingeniería y Matemáticas existen problemas que son intratables y que hasta hace algunos años, desde el punto de vista de la teoría de la complejidad computacional habían sido abordados desde diferentes enfoques como algoritmos exactos, heurísticas, algoritmos paralelos, algoritmos de aproximación, entre otros.

Recientemente este tipo de problemas, también se han explorado, utilizando el enfoque de complejidad parametrizada. La idea fundamental de este enfoque consiste en restringir la explosión combinatoria que es "inevitable", la cual es responsable de provocar el crecimiento exponencial en el tiempo de ejecución de ciertos parámetros específicos del problema, como se muestra en la Figura 1.



a) b)

Figura 1. a) Explosión Combinatoria en la Complejidad Tradicional. b) Explosión Combinatoria en la Complejidad Parametrizada $f(k)n^{\alpha}$.
(tomada de Downey et al (2012))

El análisis de HRG es en general un problema intratable, sin embargo, para un tipo restringido de HRG, se pueden desarrollar algoritmos eficientes bajo un enfoque de Parámetro Fijo Tratable (FPT), por sus siglas en inglés, donde se pueden usar las métricas como ancho hiperarbóreo, ancho de corte, ancho de ruta, ancho de clique, entre otros. Las HRG son un tipo de gramáticas generales que conjuntamente con la métrica y parámetros

adecuados conduce a soluciones eficientes de problemas intratables (Downey et al, 1995 y Cygan et al, 2016).

El artículo está organizado de la siguiente manera. En la sección 2 se abordan algunos conceptos relacionados con las HRG, tales como, hipergrafos, el mecanismo de reemplazo de hiperaristas por hipergrafos y se describen las $HRGs$. También se abordan conceptos como descomposición hiperarbórea y métricas hiperarbóreas. En la sección 3 se presentan los principales avances reportados en los trabajos de investigación relacionados a las métricas de grafos e hipergrafos; por último, en las conclusiones se discuten algunas posibles líneas de investigación.

2 Preliminares y definiciones

En esta sección, se abordan algunos conceptos como hipergrafos, reemplazo de una hiperarista por un hipergrafo, elementos necesarios para abordar el concepto de la HRG. También se abordan conceptos como descomposición hiperarbórea y métricas hiperarbóreas.

2.1 Hipergrafos y Gramáticas de Reemplazo de Hiperaristas

Un hipergrafo es la generalización de un grafo, donde cada hiperarista e puede unir a cualquier subconjunto de vértices, denominados vértices adjuntos, incluyendo al conjunto vacío (Cuticapean, 2019). Un hipergrafo se forma a partir de un conjunto de vértices junto con un número variado de hiperaristas dirigidas. (Habel, 1992, Rozenberg et al, 1986 y Engelfriet 1997).

Definición 1 Un hipergrafo con hiperaristas etiquetadas y dirigidas es una terna $H = (V, E, lab)$, donde V es un conjunto de vértices, cada $e \in E$ es un par (R_e, D_e), tal que $R_e \subseteq V$ es el origen de e y $D_e \subseteq V \setminus R_e$ es el destino, C es un conjunto numerable de etiquetas y lab es una función de E en C.

Definición 2 Sea $H = (V, E, lab)$ un hipergrafo, sea V un conjunto de vértices y $a, b \in V$. Entonces a es un vértice adyacente a b si existe una hiperarista $e \in E$ tal que $a \in R_e$ y $b \in D_e$.

Los vértices externos se definen como una lista ordenada de vértices distintos $ext \in V^*$. El vértice raíz se define como un vértice designado como la raíz del hipergrafo y cada hiperarista es dirigida desde esta raíz, ambos elementos especifican cómo reemplazar una hiperarista por un hipergrafo.

Las HRG permiten manipular hipergrafos mediante la sustitución de hiperaristas. Una hiperarista $e \in H$ es reemplazada por un hipergrafo H', a través del vértice raíz y los vértices externos, inicialmente la arista e es removida y los vértices externos de H', son mapeados con los vértices adjuntos de e, es decir el i-ésimo y j-ésimo vértice externo de la hiperarista e es mapeado con el i-ésimo y j-ésimo vértice adjunto en el hipergrafo H'.

Definición 3 Lautemann (1990) Sean H_C la clase de todos los hipergrafos sobre el conjunto de etiquetas C. Sean $H \in H_C$ un hipergrafo y $B \subseteq E(H)$ un conjunto de hiperaristas a ser reemplazadas. Sea $repl: B \rightarrow H_C$ una función que realiza la operación de reemplazo de la siguiente manera.

El reemplazo de B en H hecho por $repl$ produce el hipergrafo $H[repl]$ que se obtiene al quitar B de CE_H, agregando disjuntamente los vértices e hiperaristas de $repl(e)$ por cada $e \in B$ y empatando el i-ésimo vértice externo con el i-ésimo vértice adjunto de e para cada $e \in B$. Todas las hiperaristas mantienen sus etiquetas y vértices adyacentes; los vértices externos de $H[repl]$ son los de H. Si $B = \{e_1, \dots, e_n\}$, y $repl(e_i) = R_i$, para $i = 1, \dots, n$, entonces se escribe $H\left[{}^{e_1}/_{R_1}, \dots, {}^{e_n}/_{R_n}\right]$ en lugar de $H[repl]$.

2.2 Gramáticas de Reemplazo de Hiperaristas

Las HRG son elementos que pueden ser empleados para la generación y análisis de la representación semántica apoyada en hipergrafos (Groschwitz et al, 2015). En la Figura 2 inciso a) se muestra una HRG.

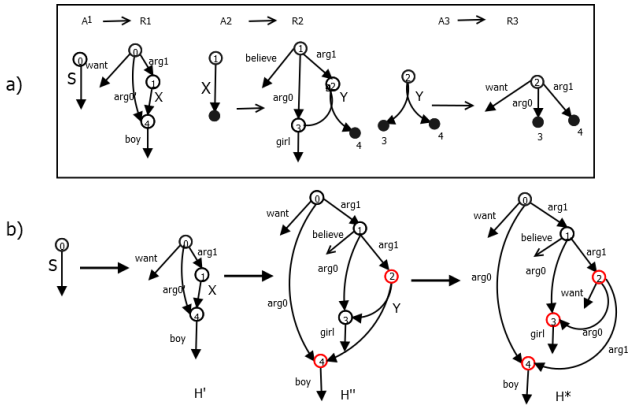


Figura 2. a) Reglas de producción para G. b) Derivación a partir de G para hipergrafo H^* que se representa el significado "The boy wants the girl to believe that he wants her". (tomada de Chiang et al (2013))

Definición 4 Peng et al (2015) Una Gramática de Reemplazo de Hiperaristas (HRG) es una tupla $G = (N, T, P, S)$ donde:
— N es un conjunto finito de símbolos no terminales.
— T es un conjunto finito de símbolos terminales.
— $S \in N$ es un símbolo no terminal inicial.
— P es un conjunto finito de reglas de producción de la forma $p: A \rightarrow R$, donde $A \in N$, y R es un hipergrafo cuyas hiperaristas están etiquetadas por símbolos de $T \cup N$.

2.3 Descomposición hiperarbórea y métricas

Los problemas que se basan en hipergrafos con ancho de hiperárbol acotado pueden ser tratables (Bodlaender, 1997, 1998 y Gildea, 2011), siempre y cuando cada propiedad de hipergrafo pueda ser definible en lógica monádica de segundo orden (Courcelle et al, 2012).

Un hiperárbol de un hipergrafo $H = (V, E)$ es una 3-tupla $\langle T, x, \lambda \rangle$, donde $T = (N, F)$ es un árbol enraizado y $x\ y\ \lambda$ son funciones etiquetadoras que asocian a cada nodo $p \in N$ con dos conjuntos: $x(p) \subseteq V$ y $\lambda(p) \subseteq E$. Denotan el subárbol enraizado al nodo $p \in N$ con T_p y sea $x(T_p) = \{v \mid v \in x(w), w \in T_p\}$ (Gottlob et al, 2009).

El ancho de corte de un grafo es el mínimo ancho de corte de todas las composiciones de vértices de $V(G)$.

Definición 5 Gottlob et al (2009). Un árbol de descomposición de un hipergrafo $H = (V, E, lab)$ es un hiperárbol $HD = \langle T, x, \lambda \rangle$, tal que cumple las siguientes condiciones:

1. Para cada hiperarista $e \in E$, hay un nodo $p \in N$, tal que $vertices(e) \subseteq x(p)$,
2. Para cada vértice $v \in V$, el conjunto $\{p \in N \mid v \in x(p)\}$ induce un subárbol conectado de T,
3. Para cada $p \in N, x(p) \subseteq vertices(\lambda(p))$,
4. Para cada $p \in N$, $vertices\ (\lambda(p)) \cap x(T_p) \subseteq x(p)$.

El ancho de un árbol de descomposición hiperarbórea T es $width(T) = máx_{t \in T} \mid \lambda(t) \mid -1$

Sea T_H la familia de todos los árboles de descomposición de H, se define el ancho del hipergrafo H como el $widt(H) = mín_{t \in T_H}\{width(T)\}$.

Definición 6 (Hamm, 2019) Un arreglo lineal de un hipergrafo H es una biyección $\varphi : V(H) \leftrightarrow \{1, \dots, |V(H)|\}$. El ancho de corte de un arreglo lineal φ de la posición $i \in \mathbb{N}$ es definido como $cw(\varphi, i) = |\{e \in E(H) \mid \exists\, v, w \in e\ \varphi(v) \leq i < \varphi(w)\}|$. El ancho de corte de un arreglo lineal φ es definido como $cw(\varphi) = max_{i \in \mathbb{N}} cw(\varphi, i)$. El ancho de corte de un hipergrafo H está definido como $cw(H) = min_{arreglo\ lineal\ \varphi\ de\ H}\ cw(\varphi)$.

2.4 Complejidad Parametrizada

El concepto de *FPT* permite hacer tratables a los problemas que son intratables, gracias a que se puede identificar y fijar alguna restricción, es decir identificar el parámetro en algún valor constante que lleve a mejorar el comportamiento computacional de los algoritmos que resuelven dichos problemas intratables (Downey et al, 1995b).

Definición 7 (Downey et al, 1995b). Un problema $L \subseteq \sum* \ X \ \sum*$ es un parámetro fijo tratable si hay un algoritmo que de forma correcta decide, para la entrada $(x, y) \in \sum* \ X \ \sum*$ ya sea $(x, y) \in L$ en tiempo $f(k)n^{\propto}$, donde n es el tamaño de la parte principal de la entrada x, $\|x\| = n$, k es el parámetro que podemos tomar para que sea de longitud de y, $k = \|y\|$, $\propto$ es una constante (independiente de k), y f es una función arbitraria.

3 Trabajo Relacionado

Se muestra que el lenguaje generado por una gramática libre de contexto siempre es acotado por el ancho del árbol (Blume et al, 2013).

El interés por la relación entre las descomposiciones de árboles y la reescritura de hipergrafos parece haber disminuido, sin embargo, es un tema que tiene mucho potencial para seguir explorando la interacción entre la transformación de hipergrafos y la teoría de grafos, ya que las descomposiciones de hipergrafos y métricas de anchura siguen siendo de interés para la comunidad de la teoría de grafos (Blume et al, 2013).

3.1 Ancho de árbol

Existen problemas clásicos que son intratables (Chandra et al, 1997), como bases de datos y satisfacción de restricciones. Sin embargo, una de las líneas de investigación para dar solución a estos problemas a través de un enfoque natural, el cual se basa en la búsqueda de propiedades en la estructura del hipergrafo subyacente, de tal forma que se asegura una solución tratable para dichos problemas (Yannakakis, 1981).

La utilización de las propiedades estructurales de las instancias del problema, cuyo hipergrafo subyacente es acíclico, son el punto clave para la tratabilidad de problemas que de otro modo serían intratables; tales como, satisfacción de restricciones, problemas que surgen en las aplicaciones de Inteligencia Artificial (IA), entre otros (Yannakakis, 1981 y Dechter, 1999). Sin embargo, como el orden del polinomio depende del ancho, el cual es una métrica inevitable; por lo tanto, no es FPT, parametrizado por cualquiera de sus métricas de ancho (Ganian et al, 2020).

La descomposición hiperarbórea juega un rol similar para hipergrafos así como el árbol de descomposición para grafos. El ancho arbóreo es el número mínimo de hiperaristas necesarias para cubrir todas las bolsas de un árbol de descomposición Gottlob et al (2003) y Addler et al (2007) presentan los conceptos de árbol de descomposición y ancho de árbol de un hypergrafo aplicados a los hipergrafos de dependencia para sistemas dinámicos.

Chiang et al (2013), propone un algoritmo para el análisis de HRG pero no lo presenta formalmente como una solución FPT. Sin embargo, se puede formalizar la parametrización de su algoritmo, a través del ancho de hiperarbóreo del hipergrafo; para lo cual, se utiliza la técnica denominada acotación arbórea junto con la técnica de programación dinámica.

Una de las métricas de aciclicidad asociada al árbol de descomposición de un hipergrafo, es el ancho hiperarbóreo. El árbol de descomposición y el ancho de árbol son conceptos que pueden ser generalizados de forma natural a hipergrafos; Intuitivamente, éste mide que tan acíclico es un hipergrafo. El ancho hiperarbóreo es igual al ancho de árbol de su grafo primario (Ganian et al, 2020).

Por lo que sabemos, ha habido pocas investigaciones sobre las nociones de ancho de ruta y ancho de árbol en el contexto de la reescritura de grafos. Sin embargo, en (Fischl, et al 2019, Gottlob, 2020 y Ganian et al, 2020) analizan la relación entre gramáticas libres de contexto (o reemplazo de hiperaristas) y ancho de árbol acotado.

3.2 Ancho de corte

El ancho de corte de un grafo G es el entero k más pequeño tal que los vértices de G se pueden ordenar en una composición lineal $[v_1, \dots, v_n]$ tal que, para cada $i = 1, \dots, n-1$, hay a lo más k aristas con un punto final en $\{, \dots, v_i\}$ y el otro punto en $\{v_{i+1}, \dots, v_n\}$.

El ancho de corte es una métrica para grafos que se puede generalizar de forma natural a hipergrafos y que ha sido estudiado para diferentes aplicaciones.

(Thilikos et al, 2005) propone un algoritmo en tiempo lineal que determina, si G tiene un ancho de corte de a lo más k, para cualquier grafo G de entrada y cualquier constante k fija.

El algoritmo propuesto por Miller et al (1991) no es FPT, ya que es del $O(n^m)$, (Hamm, 2019). Para encontrar un orden con el menor ancho de corte posible, es decir, resolver el problema de orden lineal de corte mínimo se sabe que es un problema intratable (Makedon et al, 1983).

Sin embargo, si se considera el parámetro k acotado para el ancho de corte máximo permitido, entonces se puede decidir en tiempo lineal sí un orden lineal de un hipergrafo es posible sin que se exceda el límite. (Makedon et al, 1983) propone un algoritmo FPT parametrizado por el grado máximo del vértice y el ancho de ruta de su grafo de incidencia.

Hamm (2019) muestra que existe un algoritmo de tiempo lineal que para cualquier hipergrafo H, decide si H tiene un ancho de corte acotado por una constante k fija. También muestra que el ancho de corte de un hipergrafo es igual al producto de su máximo grado de vértice y el ancho de ruta de su grafo de incidencia.

3.3 Ancho de ruta

Intuitivamente, en la teoría de grafos, una descomposición de la ruta de un grafo G es, una representación de G como una ruta de grafo "engrosado", y el ancho de ruta de G es un número que mide cuánto se engrosó la ruta para formar al grafo G (Diestel et al, 2005), es decir el ancho de ruta es una métrica de similitud del hypergrafo a un camino, es decir que tan ancha es la ruta (Makedon et al, 1983).

Más formalmente, una descomposición de ruta es una secuencia de subconjuntos de vértices de G tal que los extremos de cada arista aparecen en uno de los subconjuntos y cada vértice aparece en una subsecuencia contigua de los subconjuntos, (Robertson et al, 1983) y el ancho de ruta, es uno menos que el tamaño del conjunto más grande en tal descomposición.

Una observación clave es que, si se fija una cota, entonces también se puede limitar el ancho de ruta del grafo de incidencia de un hipergrafo (Hamm, 2019) de tal forma que, si se usa el resultado reportado por Bodlaender et al, (1996b), es posible calcular la descomposición de ruta correspondiente al grafo de incidencia en un tiempo lineal.

Obtener la descomposición de ruta, así como el ancho de ruta, no es un problema trivial, sin embargo, ha sido estudiado por (Bodlaender, 1996a y Bodlaender et al, 1996b). Muestra

que para un parámetro k fijo se puede comprobar en tiempo lineal (el tamaño del grafo) que el grafo tiene un ancho de ruta y también se puede obtener la descomposición de camino.

Existe una relación directa entre el ancho de ruta de un grafo y la descomposición de camino, de forma más general con el árbol de descomposición; de hecho, el ancho de ruta se considera como un caso especial del ancho de árbol. En el área de Procesamiento del Lenguaje Natural (PLN) se utilizan grafos que tienen un ancho de ruta pequeño (Niedermeir, 2006). El ancho de ruta del grafo de incidencia para un grafo es acotado por el ancho de corte de un hipergrafo (Hamm, 2019).

3.4 Ancho de clique

La noción de ancho de clique se introdujo a principios de la década de los 1990s (Courcelle et al, 1993).

El concepto de ancho de clique como una medida para el estudio de la complejidad de problemas que se basan en grafos y que son intratables, es asociado con descomposiciones jerárquicas e introducida por Courcelle et al (2000). Muchos de los problemas intratables, tienen algoritmos de complejidad lineal en grafos con un árbol de descomposición de ancho acotado, por algún k fija y lo mismo sucederá para grafos con ancho de clique de a lo más k. Courcelle et all (2000) muestran que cualquier grafo de ancho de árbol acotado, también es de ancho de clique acotado.

El ancho de clique de un hypergrafo H puede ser definido como el ancho correspondiente al grafo de incidencia o al grafo primal y, además, el ancho de clique de su grafo primal es acotado por alguna constante k fija.

Las restricciones en las estructuras en términos del ancho de árbol y ancho de clique han sido estudiadas en el área de gramáticas de hipergrafos. De hecho, las restricciones del ancho de clique permiten una clase de estructuras para grafos mucho más grande que la que permite el ancho de árbol (Gottlob et al, 2004).

Courcelle et all (2000) En general, cada problema de grafos que puede ser expresado en Lógica Monódica de Segundo Orden con cuantificaciones sobre los vértices y conjuntos de éstos, pueden ser resueltos en tiempo lineal, si la entrada del grafo es dada con una $k - expresión$.

Sang (2008) existen problemas de grafos intratables pero que pueden ser resueltos en tiempo polinomial siempre y cuando una descomposición arbórea correspondiente al ancho de clique de a lo más k, llamada $k - expresión$, sea dada como entrada junto con la lista de adyacencias del grafo de entrada. De hecho, el ancho de clique es el mínimo k que hay en una $k - expresión$ de G.

Para cada grafo G con ancho de árbol de a lo más k tiene un ancho de clique de a lo más $O(2^k)$. El ancho clique es una buena medida de la complejidad de grafos densos y que al ser acotado en el grafo de incidencias implica un ancho de hiperárbol acotado (Gottlob et al, 2004). Por lo tanto, un gran número de problemas intratables en general se pueden hacer tratables en instancias de ancho de clique acotado. Sin embargo, generalizar el ancho de

clique a hipergrafos uniformes no es una tarea simple (Stozecki, 2012, Courcelle, 2000 y Addler et al, 2008).

La importancia de éstas métricas de ancho, para grafos, es por el hecho de que existen problemas que en general son intratables. Sin embargo, se pueden solucionar en tiempo polinomial cuando la entrada se restringe a grafos con ancho de clique acotado. (Courcelle, 2000 y Kamiński et al, 2009).

Métrica	Grafos	Parametrización para grafos	Hipergrafos	Parametrización para hipergrafos
Ancho de Árbol	si	si	si	si
Ancho de Corte	si	si	si	si
Ancho de Ruta	si	si	no	no
Ancho de Clique	si	si	no	no

Tabla 1. Métricas y parametrización en Hipergrafos.

Existen diferentes métricas utilizados en el análisis de la complejidad parametrizada, tales como ancho de árbol, ancho de ruta, ancho de corte, ancho de clique, etc. Son algunos ejemplos de la variedad de posibles métricas. De hecho, (Marx et al, 2021) presentan dos algoritmos, en los que parametrizan por ancho de árbol y ancho de corte.

En la Tabla 1, se muestra que, en la bibliografía explorada, se reportaron algunos trabajos que hacen uso de las métricas de ancho de árbol y de corte en soluciones parametrizadas para grafos e hipergrafos, lo cual no se ve reflejado para el ancho de ruta y ancho de clique.

Debido a lo anterior, se observa que para las métricas ancho de árbol y el ancho de corte, se han presentado algunos trabajos para hipergrafos, lo que motiva a una exploración más exhaustiva y en todo caso su aplicación en trabajos subsecuentes que se enfoquen a la solución eficiente de algunos problemas intratables.

4 Conclusiones

En vista de la revisión realizada para algunas métricas, se puede concluir que, gracias a que se han reportado resultados de parametrización en grafos y que además pueden ser generalizados a hipergrafos, el uso de algunas de las métricas exploradas, como se muestra en la Tabla 1, pueden conducir a una solución FPT para el análisis de las HRG.

4.1 Trabajo a futuro

Como trabajo a futuro y en base a la presente revisión, se puede mencionar la necesidad de realizar un trabajo más exhaustivo para explorar el uso de otras métricas arbóreas que conduzcan a algoritmos eficientes para algunos problemas hasta ahora intratables, al mismo

tiempo que puedan ser generalizadas a hipergrafos y que se ajusten a los conceptos relacionados al tema de FPT para el análisis de las $HRGs$.

Debido a la extensión de los GGQ (Consulta de Grafos Generalizada) hacia GHQ (Consulta de Hipergrafos Generalizada) es de manera natural, estas métricas pueden ser utilizadas para el análisis de los árboles de decisión de un conjunto de objetos de entrenamiento, descritos a través de una colección de propiedades.

También se realizará una revisión de la descomposición por relleno para grafos y sus métricas de árbol, tales como árboles embebidos y longitud de árbol acotado que presenta (Chekuri, 2009) para algoritmos de aproximación, con el afán de explorar la viabilidad a ser generalizadas a hipergrafos y ajustadas al concepto de FPT.

Referencias

Addler, I., Gottlob, G., Grohe, M. (2007). "Hypertree width and related hypergraph invariants", *European Journal of Combinatorics*, vol. 28, pp. 2167-2181.

Addler, H., Addler, I. (2008). "A note on clique-width and tree-width for structures", *Journal ArXiv*, Recuperado de: https://arxiv.org/abs/0806.0103.

Blume, C., Bruggink, H.J., Friedrich, M., König, B. (2013). "Treewidth, pathwidth and cospan decompositions with applications to graph-accepting tree automata", *Journal of Visual Languages and Computing*, vol. 24, pp. 192-206.

Bodlaender, H.L. (1996a). "A Linear-Time Algorithm for Finding Tree-Decompositions of Small Treewidth", *SIAM Journal on Computing*, vol. 25, pp. 1305-1317.

Bodlaender, H.L., Kloks, Ton. (1996b). "Efficient and Constructive Algorithms for the Pathwidth and Treewidth of Graphs", *Journal of Algorithms*, vol. 21, pp. 358-402.

Bodlaender, H.L. (1997). "Treewidth: Algorithmic techniques and results", *International Symposium on Mathematical Foundations of Computer Science*, vol. 1295, pp. 9–36, Springer, Berlin, Heidelberg.

Bodlaender, H.L. (1998). "A Partial K-Arboretum of Graphs With Bounded Treewidth", *Theorical Computer Science*, vol. 209, pp. 1–16, Springer.

Chandra, A., Merlin, P. (1997). "Optimal implementation of conjunctive queries in relational data bases", *Proceedings of the ninth annual ACM symposium on Theory of computing*, pp. 77-90.

Chekuri, S. (2009). https://courses.engr.illinois.edu/cs598csc/sp2009/Lectures/lecture_23_draft.pdf

Chiang, D., Andreas, J., Bauer, D., Hermann, K.M., Jones, B., Knight, K. (2013). "Parsing Graphs with Hyperedge Replacement Grammars", *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, vol. 1. pp. 77-90.

Courcelle, P.B., Olariu, S. (2000). "Upper bounds to the clique width of graphs", *Discrete Applied Mathematics*, vol. 101, pp. 924-932.

Courcelle, P.B., Engelfriet, D.J. (2012). "Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach", *Encyclopedia of Mathematics and its Applications*, pp. 505-577, Cambridge University Press.

Cygan, M., Fomin, F. V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., y Saurabh, S. (2016). "Parameterized algorithms" en Cygan, M., Fomin, FV., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Saurabh, S. (eds), *Computer Science,* (1st ed., pp. 1-613), Springer, Cham.

Curticapean, R. (2019). “Counting Problems in Parameterized Complexity” en Christophe, P. y Pilipczuk, M. (ed), *13th International Symposium on Parameterized and Exact Computation (IPEC 2018),* (vol. 115, pp. 1-18), Schloss Dagstuhl--Leibniz-Zentrum fuer Informatik.

Dechter, R. (1999). “Bucket elimination: A unifying framework for reasoning, Artificial Intelligence”, *Artificial Intelligence*, vol. 113, pp. 41-85.

Diestel, R., Kuhn, D. (2005). “Graph Minors Hierarchies”, *Artificial Intelligence*, vol. 113, pp. 41-85.

Downey, R.G., Fellows, M.R. (1995). “Discrete Applied Mathematics”, vol. 145, pp. 167-182.

Downey, R.G., Fellows, M.R. (1995b). “Parameterized computational feasibility, in feasible mathematics”, vol. pp. 219-244.

Downey, R.G., Fellows, M.R. (2012). “Parameterized Complexity”. *Springer Science & Business Media*.

Drews, F., Kreowski, H., y Annegret, H. (1997). “Hyperedge replacement graph grammars” en Grzegorz, R. (ed), *Handbook of graph grammars and computing by graph transformation,* (vol. 1, pp. 95-162), World Scientific.

Engelfriet, J. (1997). “Context-Free Graph Grammars” en Rozenberg, G., Salomaa, A. (eds), *Handbook of Formal Languages,* (vol. 3, pp. 125-213), Springer Berlin Heidelberg.

Fischl, W. Gottlob, G., Pichler, R. (2019). “General and Fractional Hypertree Decompositions: Hard and Easy Cases”, *SIGMOD/PODS '18: Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database System*, vol. 1, pp. 17-32.

Gallo, G. Scutella, M., (1998). “Directed hypergraphs as a modelling paradigm”, *Decisions in Economics and Finance*, vol. 21, pp. 97-123 doi: 10.1007/BF02735318.

Ganian, R., Schidler, A., Sorge, M., y Szeider, S. (2020). “Threshold Treewidth and Hypertree Width” en Bessiere, C. (ed), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, (IJCAI-20),* pp. 1898-1904, International Joint Conferences on Artificial Intelligence Organization.

Gildea, D. (2011). “Grammar Factorization by Tree Decomposition”, *Comput. Linguist*, vol. 37, pp. 231-248.

Gottlob, G., Leone, N., y Scarcello, F. (2002). “Hypertree Decompositions and Tractable Queries”, *Journal of Computer and System Sciences*, vol. 64, pp. 579-627.

Gottlob, G., Leone, N., y Scarcello, F. (2003). “Robbers, marshals, and guards: game theoretic and logical characterizations of hypertree width”, *Journal of Computer and System Sciences*, vol. 66, pp. 775-808.

Gottlob, G., Pichler, R. (2004). “Hypergraphs in Model Checking: Acyclicity and Hypertree-Width versus Clique-Width”, *Society for Industrial and Applied Mathematics*, vol. 33, pp. 351-378. SIAM J, Comput.

Gottlob, G., Miklos, Z., Schwentick, T. (2009). “Generalized Hypertree Decompositions: NP-Hardness and Tractable Variants”, *Journal of ACM*, vol. 56, pp. 1-32.

Gottlob, G., Greco, G., y Scarcello, F. (2014). “Treewidth and hypertree width”, *Tractability: Practical Approaches to Hard Problems*, pp. 3-38.

Gottlob, G. (2020). “Complexity Analysis of Generalized and Fractional Hypertree Decompositions”, *Journal Journal of the (ACM)*, vol. 68, pp. 1-50.

Groschwitz, J., Koller, A., Teichmann, C. (2015). “Graph parsing with s-graph grammars”, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015*, vol. 1, pp. 1481-1490. ACL 2015. The Association for Computert Linguistics.

Habel, A. (1992). “Introduction to hyperedge-replacement grammars” en Habel, A. (ed), *Hyperedge Replacement: Grammars and Languages,* (1st ed., pp. 43-53), Springer Book Archive.

Hamm, T. (2019). “Finding Linear Arrangements of Hypergraphs with Bounded Cutwidth in Linear Time” en Jansen, B. y Telle, JA. (ed), *14th International Symposium on Parameterized and Exact Computation (IPEC2019),* (vol. 148, pp. 1-14), Schloss Dagstuhl--Leibniz-Zentrum fuer Informatik.

Huang, J., Zhang R., y Yu, J.X. (2015). “Scalable Hypergraph Learning and Processing”, *2015 IEEE International Conference on Data Mining*, pp. 775-780, doi: 10.1109/ICDM.2015.33.

Kamiński, M., Lozin V. V., Milanič, M. (2009). “Recent developments on graphs of bounded clique-width, Discrete Applied Mathematics”, *Discrete Applied Mathematics*, vol. 157, pp. 2747-2761.

Lautemann, C. (1990). “The complexity of graph languages generated by hyperedge replacement”, *Acta Informática*, vol. 27, pp. 399-421 doi: 10.1007/BF00289017.

Makedon, F.S., Papadimitriou, C.H., Sudborough, I.H. (1983). “Topological bandwidth” en Ausiello, G., Protasi, M. (eds), CAAP*'83. CAAP 1983. Lecture Notes in Computer Science,* (vol. 159, pp. 317-331), Springer, Berlin, Heidelberg.

Marx, D., Sankar, G.S., Schepper, P. (2021). “Degrees and Gaps: Tight Complexity Results of General Factor Problems Parameterized by Treewidth and Cutwidth, *ICALP*, vol. Abs/2105.08980, pp. 11-40 Journal (CoRR).

Miller, Z., Sudborough, I.H. (1991). “A polynomial algorithm for recognizing bounded cutwidth in hypergraphs”, *Mathematical systems theory*, vol. 24, pp. 11-40.

Moyano, F.J.M.R., Gómez, A., Walther, D. (2016). “Strong Connectivity in Directed Hypergraphs and its Application to the Atomic Decomposition of Ontologies”, *Tesis Doctoral*, E.T.S. de Ingenieros Informáticos (UPM), Inteligencia Artificial.

Niedermeier, R. (2006). “Invitation to Fixed-Parameter Algorithms”, *Journal Invitation to Fixed-Parameter Algorithms*.

Peng, X., Song, L., y Gildea, D. (2015). “A Synchronous Hyperedge Replacement Grammar based approach for AMR parsing”, *19th Conference on Computational Language Learning*, pp. 32-41, Association for Computational Linguistics.

Peuser, C., (2018). “From Hyperedge Replacement Grammars to Decidable Hyperedge Replacement Games” en Mazzara, M., Ober, I., Salaün, G. (eds), *Software Technologies: Applications and Foundations,* (vol. 11176, pp. 463-478), Springer, Cham.

Robertson, N., Seymour, P. (1983). “Graph minors. I. Excluding a forest”, Journal of Combinatorial Theory, vol. 35, pp. 39-61.

Rozenberg, G., Welzl, E. (1986). “Boundary NLC graph grammars—Basic definitions, normal forms, and complexity”, *Journal Information and Control*, vol. 69, pp. 136-167.

Sang, O. (2008). “Approximating Rank-Width and Clique-Width Quickly”, *Association for Computing Machinery*, vol. 5, pp. 1-20.

Stozecki, Y. (2012). “Decomposition-width: Extending the Clique-width to Hypergraphs”. Recuperado de: https://yann-strozecki.github.io/hypergraph_decomposition.pdf.

Thilikos, D., Serna, M., Bodlaender, H. (2005). “Cutwidth I: A linear time fixed parameter algorithm”, *Proceedings of the Seventh International Conference on Very Large Data Bases*, vol 56, pp. 1-24, VLDB Endowment.

Yannakakis, M. (1981). “Algorithms for Acyclic Database Schemes”, *Proceedings of the Seventh Journal of Algorithms*, vol 7, pp. 82-94.

Capítulo 8

Conteo de conjuntos independientes sobre un grafo tipo malla

Guillermo De Ita Luna, Luis Filiberto Regino Medina, Beatríz Bernabé Loranca

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

deitaluna63@gmail.com, luis.regino@alumno.buap.mx,
beatriz.bernabe@gmail.com

Resumen. El problema de conteo de números independientes de un grafo G no es solamente matemáticamente relevante e interesante, sino también tiene muchas aplicaciones; en física, matemática o en la ciencia teórica de la computación.

En este artículo se presenta un método novedoso para el conteo de conjuntos independientes sobre estructuras tipo malla. Se parte de explicar las recurrencias que usa el método para contar conjuntos independientes sobre topologías básicas de grafos. El método se extiende para procesar estructuras tipo mallas de caras cuadráticas. La propuesta tiene una complejidad en tiempo mucho menor a la que requiere el método líder y actual basado en la matríz de transferencia, para el conteo de conjuntos independientes sobre mallas.

Palabras Clave: Conteo de conjuntos independientes, Grafo tipo malla, Matriz de transferencia, Recurrencia Fibonacci.

1 Introducción

El conteo se ha convertido en un área importante en matemáticas, así como en las ciencias de la computación, a pesar de que ha recibido una menor atención que los problemas de decisión. Esto ha provocado que se tenga menos conocimiento sobre la complejidad de problemas de conteo comparada con el estudio de la complejidad sobre problemas de decisión.

En un ámbito computacional, el conteo de conjuntos independientes de un grafo es un factor determinante para establecer la frontera entre conteo eficiente y procedimientos de conteo intratables. En la actualidad, son muy pocos los problemas de conteo de grafos que pueden ser resueltos en tiempo polinomial.

Se muestra en (Vadhan, 2001) que el conteo de conjuntos independientes (CCI) en grafos de grado 4 es un problema de la clase de complejidad $\#P-$ Completo. Greenhill (Greenhill, 2000) refinó este trabajo demostrando que el conteo de conjuntos independientes de un grafo de grado 3 o grafos 3 regulares es de igual manera $\#P$-Completo. Una importante línea de investigación es determinar el tipo de grafos donde el conteo de conjuntos independientes

pueda realizarse en tiempo polinomial.

Las cadenas de Markov descrita en (Luby y Vidoga, 1997) es una de las primeras aproximaciones de algoritmos de conteo de conjuntos independientes. La aplicación del algoritmo de la cadena de Markov en un sistema Monte Carlo ha alcanzado una buena aproximación, en tiempo polinomial, del número de conjuntos independientes en un grafo G, en especial para grafos con un máximo de grado cuatro (Dyer, 1997). Se han desarrollado muchas variantes del algoritmo Monte Carlo, para mayor detalle ver [(Luby, 1997), (Dyer, 1997, 2002, 2004), (Russ, 2001)], estas técnicas de aproximación fallan en grafos con grado 6 o mayor, y el caso de grafos de grado 5, el problema permanece abierto (Dyer, 1997).

Hay una gran cantidad de literatura enfocada al conteo de estructuras sobre grafos tipo malla, como por ejemplo contar arboles de expansión, ciclos hamiltoneanos, conjuntos independientes, u orientaciones acíclicas, así como conteo de coloreos [(Calkin,1998), (Euler, 2005, (Mordecai,2005), (Guillen, 2008)].

Dahllöf (2002) diseñó un método para el conteo de modelos en fórmulas Booleanas (que serviría para contar conjuntos independientes sobre fórmulas monótonas), y cuyo algoritmo está acotado superiormente en el peor de los casos por $O(1.3247^n)$, siendo n el número de variables (vértices) de la fórmula. Mientras en (Okamoto, 2005) se desarrolló un algoritmo de tiempo lineal para el conteo de conjuntos independientes para grafos chordales.

En (Calkin, 1998) se calcula el número de conjuntos independientes de un grafo tipo malla *Gm,n* (*m* renglones y *n* columnas, ver figura 1 que ilustra una malla de 4 x 6), utilizando el método de la matriz de transferencia. Mientras que Euler (Euler, 2005) presentó las funciones generatrices asociadas al conteo del número de conjuntos independientes sobre la malla. Euler también consideró el conteo de conjuntos independientes maximales en una malla. Sin embargo, la aplicación del método de la matriz de transferencia para contar el número de conjuntos independientes en una malla *Gm,n,* tiene un carácter de complejidad exponencial en tiempo sobre ambas dimensiones (*m* y *n*).

En particular, el conteo de conjuntos independientes en mallas está relacionado con los modelos "hard-square" usados en física estadística, y tiene un interés particular para calcular el factor de entropía constante "hard square" de un sistema físico. En mecánica estadística, contar conjuntos independientes se interpreta como contar el número de diferentes formas de poner partículas en retículas de cuadrados, tal que dos partículas no puedan estar en el mismo sitio o en puntos adyacentes de los cuadrados (Zhan, 2014). Otro tipo de aplicaciones del conteo de conjuntos independientes en mallas tiene que ver con esquemas de codificación eficientes para el almacenamiento de datos (Roth, 2001).

En este trabajo se explicará un método para el conteo de conjuntos independientes. Nuestra propuesta reduce dramaticamente la complejidad en tiempo que se requiere para contar conjuntos independientes sobre mallas con respecto al método clásico de la matriz de transferencia.

En el capítulo 2 se presenta la notación a usar, en el 3 se presentan topologías básicas de grafos (caminos, ciclos y árboles) donde el CCI se realiza eficientemente. En el capítulo 4 se

presenta nuestro método para poder CCI sobre estructuras tipo malla *Gm,n*. Y en el capítulo 5 se presentan las conclusiones del trabajo.

2 Notación

Sea $G = (V, E)$ un grafo no dirigido con un conjunto de vértices V y un conjunto de aristas E. Dos vértices v y w son adyacentes si existe un vértice $\{v, w\} \in E$ que los conecta. Algunas veces denotaremos una arista $\{v, w\} \in E$ como *vw*.

El vecindario para $x \in V$ es $N(x) = \{y \in V: \{x, y\} \in E\}$ y su vecindario cerrado es $N(x) \cup \{x\}$ el cual se denota como $N[x]$. La cardinalidad de un conjunto A se denota como $|A|$. El grado de un vértice x se denota como $\delta(x)$, que es $|N(x)|$, y el grado de G es $\Delta(G) = max\{\delta(x): \text{x} \in \text{V}\}$. El tamaño del vecindario de x, $\delta(N(x))$, es $\delta\big(N(x)\big) = \sum_{y \in N(x)} \delta y$.

Un camino de v a w se conforma por una secuencia de aristas: $v_0v_1, v_1v_2, \dots, v_{n-1}\, v_n$ donde $v = v_0$ y $w = v_n$ y v_k es adyacente a v_{k+1}, para $0 \leq k < n$. La longitud del camino es el número de aristas (n). Un camino se considera simple cuando $v_0, v_1, \dots, v_{n-1}, v_n$ son distintos. Un ciclo es un camino no vacío cuyo primer vértice y el último son los mismo y un ciclo simple es aquel ciclo que ningún vértice se repite con la excepción del primero y último vértice. Un grafo G es acíclico si no tiene ciclos. Un grafo de camino, un ciclo simple, y un grafo completo de *n* vértices se denotan de la siguiente forma: $P_n,\ C_n\ y\ K_n$ respectivamente.

Para un grafo $G = (V, E), S \subseteq V$ es un conjunto independiente de G si por cada dos vértices v_1, v_2 en S, $\{v_1, v_2\} \notin E$. $I(G)$ denota al conjunto de todos los conjuntos independientes de G. Un conjunto independiente $S \in I(G)$ es "maximal" si no es un subconjunto de un conjunto independiente más grande, y es "máximo" si tiene el tamaño más grande dentro de todos los conjuntos independientes en $I(G)$.

El problema de conteo de conjuntos independientes denotado como $i(G)$, consiste en contar los conjuntos independientes del grafo G. $i(G)$ es un problema $\#P$ completo para grafos tal que $\Delta(G) \geq 3$. Existen diversos procedimientos polinomiales para computar $i(G)$ cuando $\Delta(G) \leq 2$ [(Russ, 2001), (Dahllöf, 2002), (Roth, 1996)]. Todos ellos son métodos de complejidad lineal con respecto al tiempo. A continuación se presenta un procedimiento eficiente para calcular $i(G)$, en base a la topología del grafo G.

3 Topologías básicas para el conteo eficiente de conjuntos independientes.

$i(G) = \prod_{I=1}^{k} i(G)$ donde $G_i, i = 1,2, \dots, k$ son las componentes conectadas de G(Calkin, 1998). La complejidad total del tiempo de cómputo se expresa por $T\big(NI(G)\big)$, y se calcula en base a la regla del máximo: $T\big(i(G)\big) = max\big\{T\big(i(G_i)\big): G_i\ es\ un\ componentes\ conectado\ de\ G\big\}$.

Caso A

Se considerará un grafo $G = (V, E)$ el cual consiste de una simple secuencia de vértices (camino), $V = \{1, \dots, n\}$ y existen aristas tal que $a_i = \{i, i+1\}, i = 1, \dots, n-1$ para cada par de nodos secuenciales.

A los vértices $v_i \in V$ se les asocia un par (α_i, β_i) donde α_i expresa el número de conjuntos en $I(G_i)$ donde el nodo v_i no aparece y β_i el número de conjuntos en $I(G_i)$ donde el nodo v_i aparece, de esta forma: $i(G_i) = \alpha_i + \beta_i$.

El primer par (α_1, β_1) es $(1,1)$ ya que para el subgrafo inducido $G_1 = \{v_1\}, I(G_1) = \{\emptyset, \{v_1\}\}$. Si se sabe el valor para (α_i, β_i) para cada $i < n$, y como el siguiente subgrafo inducido G_{i+1} está formado por G_i añadiendo un vértice v_{i+1} y las aristas $\{v_i, v_{i+1}\}$, de esta forma es fácil visualizar que el par $(\alpha_{i+1}, \beta_{i+1})$ se construye de (α_i, β_i) aplicando la ecuación de recurrencia, a la que llamaremos recurrencia Fibonacci:

$$\alpha_{i+1} = \alpha_i + \beta_i \quad ; \quad \beta_{i+1} = \alpha_i \tag{1}$$

La serie $(\alpha_i, \beta_i), i = 1, \dots, n$ basada en la aplicación de la recurrencia (1), permite calcular $i(G_i) = \alpha_i + \beta_i \; para \; i = 1, \dots, n$. Entonces, el cálculo de $i(G)$ está basado en el cálculo incremental de $i(G_i), \; i = 1, \dots, n$. Si se realiza una búsqueda lineal sobre un grafo secuencial G iniciando en un extremo v_1 y moviéndose a sus vértices incidentes mientras se aplica la recurrencia (1), en un tiempo lineal basado en el número de vértices n, se obtiene $i(P_n) = i(G_n) = \alpha_n + \beta_n = F_{n+2}$, donde F_n es el $n - ésimo$ número de la sucesión Fibonacci.

Esto nos lleva a que si queremos realizar el proceso de conteo de conjuntos independientes sobre un camino es necesario utilizar líneas de computación. Una línea de computación es una secuencia de pares $\alpha_i + \beta_i, i = 1, \dots, n$ utilizados para calcular el número de conjuntos independientes sobre un camino con n nodos.

Caso B

En caso de tener un grafo tipo árbol se recorrerá G en base a una búsqueda a lo profundo, considerando el nodo raíz como cualquier vértice $v \in V$, y donde v será el nodo inicial de la búsqueda a lo profundo. Se denota con (α_v, β_v) al par asociado con el nodo $v (v \in G)$. Se calculará $i(G)$ mientras se realiza el recorrido en post-orden del árbol. Se tomarán las siguientes consideraciones para realizar el cálculo de $i(G)$ para un grafo tipo árbol.

- Se realiza un recorrido de *G* en post-orden.
  Cuando se visita un nodo $v \in G$ se considera las siguientes condiciones
- $(\alpha_v, \beta_v) = (1,1)$ si v es un nodo-hoja de G.
- Si v es un nodo-padre con una lista de nodos-hijos $u1, \dots, uk$, se le aplicará la recurrencia Fibonacci a todos estos nodos-hijos $(\alpha_{uj}, \beta_{uj})$ visitados con $j =$

$1, \ldots, k$ y se aplica $\alpha_v = \prod_{j=1}^{k} \alpha_{vj}$ y $\beta_v = \prod_{j=1}^{k} \beta_{vj}$. Se debe tomar en cuenta que esta consideración incluye el caso cuando v tiene sólo un hijo.

- Si v es el nodo-raíz de G entonces $(\alpha_v + \beta_v)$.

Las anteriores reglas permiten realizar el conteo de conjuntos independientes de G en tiempo $O(n+m)$ el cuál es el tiempo necesario para realizar la búsqueda en post-orden de una estructura árbol.

Caso C

Otro caso particular es cuando $G = (V, E), n = m = |V| = |E|$ es un ciclo simple. Podemos considerar el ciclo como un camino *G'* de *m* vértices más una arista adicional que liga al nodo final con el nodo inicial, arista $c_m = \{v_m, v_1\}$.

Para el conteo de conjuntos independientes sobre un ciclo simple, será necesario utilizar dos hilos o líneas de cómputo, una para el cálculo de $i(G')$ y otra para calcular $|\{S \in I(G') : v_1 \in S \wedge v_m \in S\}|$. Este cálculo puede realizarse fijando en $I(G')$ los conjuntos independientes donde aparece v_1, el cual se realiza con un hilo $(\alpha'_i, \beta'_i), i = 1, \ldots, m$ y con valores iniciales $(\alpha'_i, \beta'_i) = (0,1)$ para considerar que hay un único conjunto independiente de $I(G')$ donde aparece v_1.

Se aplica la recurrencia (1) para el cálculo de la nueva serie $(\alpha'_i, \beta'_i), i = 2, \ldots, m$ y para considerar únicamente los conjuntos independientes donde aparece v_m se considera como par final (α'_m, β'_m) solamente al valor $(0, \beta'_m)$. Para la visualización de un ejemplo revisar (Zacarias, 2017).

Lo que nos lleva a que $i(G) = i(G') - |\{S \in I(G') : v_1 \in S \wedge v_m \in S\}| = \alpha_m + \beta_m - \beta'_m = F_{m+2} - F_{m-2}$ donde F_m hace referencia al m-ésimo número de Fibonacci.

Es importante denotar que para el cierre del ciclo y del hilo tendremos la siguiente expresión:

$$\alpha_m + \beta_m - \beta'_m \qquad (2)$$

Las formulas obtenidas para los casos A y B son equivalentes a las fórmulas obtenidas por Arocha (1944) que las obtuvo usando los polinomios de Fibonacci, los cuales están definidos como $F_0(x) = 1, F_1(x) = 1$ y aplicando recursividad para $F_n(x) = F_{n-1}(x) + x \cdot F_{n-2}(x)$ para $n \geq 2$.

4. Desarrollo

Se tiene un grafo $G = (V, E)$ tipo malla de tamaño $m \times n$ con un conjunto de vértices: $V = \{(i,j) : 1 \leq i \leq m,\ 1 \leq j \leq n\}$**, y** un conjunto de artistas
$E = \{(j,i),(j+1,i))|1 \leq j < m, 1 \leq i \leq n\} \cup \{(j,i),(j,i+1))|1 \leq j < m, 1 \leq i < n\}$.
Denotaremos un grafo malla por $G_{m,n}$ donde *m* es el número de filas y *n* el número de columnas.

El caso de una malla con $m = 4$ y $n = 6$ lo podemos apreciar en la Figura 1. En la malla que se presenta, se puede notar que ya se tiene trazado el recorrido propuesto sobre los vértices de la malla. El inicio del recorrido es el vértice 11. Cuando encuentra una arista de ciclo se denota con una flecha curva, con la punta de la flecha indicando el vértice donde se cierra el ciclo. La flecha con una línea cruzada indica el fin del recorrido.

Por ejemplo, de acuerdo a la notación se infiere que la flecha curva indica que habrá un cierre de ciclo por la flecha que va de 12 a 11. Los cierres de ciclo se llevan a cabo una vez que se visiten todas las aristas del ciclo. Como se puede observar en la figura 1 el primer ciclo que se cierra es por medio de la arista $\{32,31\}$.

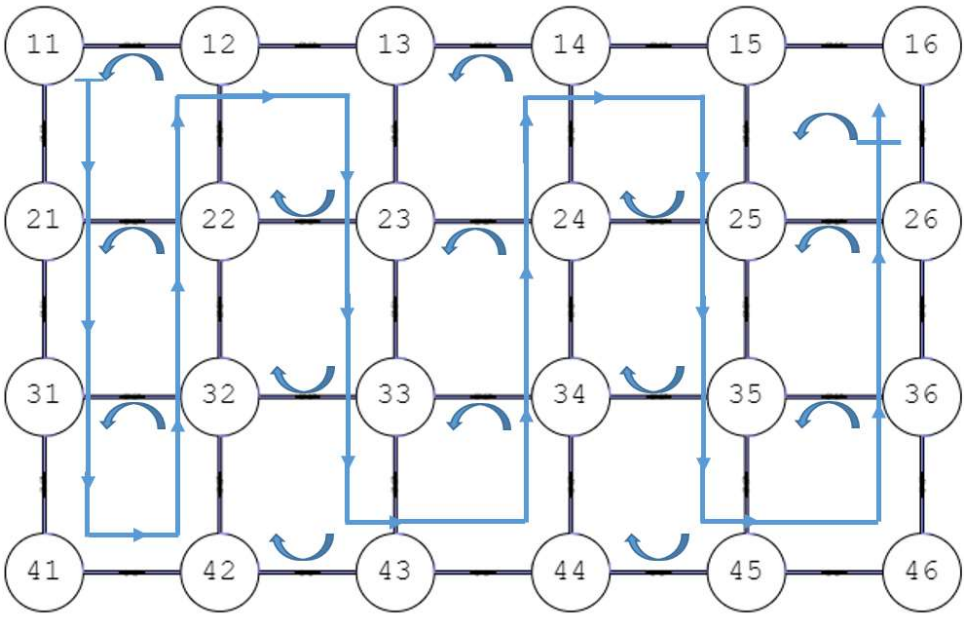


Figura 1. Grafo tipo malla y su recorrido por columnas

El recorrido inicia en el nodo 11 y cómo podemos observar, es necesario abrir una línea de computación además de la principal debido a que es necesario para cerrar más adelante el ciclo que inició en el vértice 11. Cada ciclo embebido abrirá nuevas líneas de cómputo, la cual se puede entender que sucederá durante el recorrido cuando se visitan vértices de inicio de ciclo, excepto en los vértices donde el recorrido hace un cambio de columna. Cuando se abren nuevos ciclos durante el recorrido, se crean nuevas líneas de computación, pero al mismo tiempo, se irán cerrando líneas que se habrían abierto en vértices anteriores.

En la Tabla 1 se muestra el proceso ya mencionado de abrir y cerrar líneas de computación. La Tabla 1 muestra el proceso hasta el cierre del ciclo de la arista $\{23,22\}$, que es el momento en el que el ciclo embebido alcanzará la cantidad máxima de líneas de computación abiertas. Es importante tomar las siguientes consideraciones que se realizan durante el recorrido, como la aplicación de la recurrencia, el cierre de ciclo y como se abrirán las líneas de cómputo.

- En cada movimiento de un vértice al siguiente, se aplica la recurrencia (1).
- Cada que se abre un ciclo, se abren tantas líneas de cómputo como las que estén actualmente activas con un valor $\beta \neq 0$ asociado al par (α, β) de la línea.
- Para un cierre de ciclo se aplicará la ecuación (2), donde el valor de las betas se restarán con respecto a la línea activa con la que se abrió.

Por la naturaleza sobre el crecimiento en el número de líneas de computación, la Tabla 1 se divide en subsecciones del proceso, en cada subsección se eliminarán las líneas de proceso que ya han sido cerradas, para entonces realizar la recopilación de las líneas que aún siguen abiertas. En el cierre del ciclo de la arista $\{23,22\}$ se obtiene el número máximo de líneas de computación, que puede verse que coincide con el valor de Fibonacci F_7. Se han comprobado los resultados del conteo de conjuntos independientes mostrados en la Tabla 1, usando un programa de carácter exponencial que cuenta "Conjuntos independientes" de manera exhaustiva.

La complejidad en tiempo de nuestra propuesta algorítmica está relacionado con las dimensiones de la Tabla 1, que denotaremos por $T_{k,l}$ la tabla de k filas y l columnas. Para una malla $G_{m,n}$ el número de columnas de la malla coincide con el número total de vértices que son visitados durante el recorrido lo que corresponde con el número de vértices de la malla, entonces $l = m*n$.

El número de filas k en la tabla es un valor dinámico que cambia conforme se abren y se cierran líneas de computación, y se mueven renglones en la tabla para evitar huecos en la misma. Para estimar k debemos considerar cual es el número máximo de líneas que pueden estar abiertas en cualquier momento del cálculo de $i(G_{m,n})$. El número máximo de líneas abiertas se corresponde con el número de vértices de la primera columna más un vértice de la segunda columna, que es cuando todas las líneas de ciclo están activas y no se ha cerrado ningún ciclo. Por tal, este valor corresponde con $(m+1)$ inicios de ciclo.

Cada vez que se abren líneas de ciclo se sigue un crecimiento Fibonacci. Al iniciar el recorrido se crean dos líneas de computación, luego 3, le sigue 5, 8,… que corresponde a un crecimiento Fibonacci. Así que después de $(m+1)$ inicios de ciclo se tendrían $k=F_{m+3}$ líneas activas de computación, ya que la sucesión Fibonacci inició con 2 líneas (por tal se inicia la secuencia con F_3). Por tanto, $k=F_{m+3}$.

El número máximo de celdas en la Tabla será $k * l = F_{m+3} * (m * n)$, que corresponde también con el orden de crecimiento de la complejidad en tiempo de nuestro proceso. Nuestro algotimo tiene entonces una complejidad en tiempo de orden $O(F_{m+3} * (m * n)) = O((1.618)^{m+3} * (m * n))$, tomando a 1.618 como una aproximación al 'cociente de oro' que es la constante que nos permite calcular los números de Fibonacci.

TABLA 1. CONTEO DE CONJUNTOS INDEPENDIENTES QUE CORRESPONDE A LA FIGURA 1

LÍNEAS/VÉRTICES	11	21	31	41	42	32 ↶ 31	22 ↶21	12 ↶ 11
Lp	(1,1) →	(2,1) →	(3,2) →	(5,3) →	(8,5) →	(13,8) − (0,2) = (13,6) →	(19,13) − (0,3) = (19,10) →	(29,19) − (0,7) = (29,12) →
C_{11_p}	(0,1) →	(1,0) →	(1,1) →	(2,1) →	(3,2) →	(5,3) − (0,1) = (5,2) →	(7,5) →	(12,7) →X
C_{21_P}	----------	(0,1) →	(1,0) →	(1,1) →	(2,1) →	(3,2) →	(5,3) →X	----------
C_{31_p}	----------	----------	(0,2) →	(2,0) →	(2,2) →	(4,2) →X	----------	----------
C_{31_11}	------	----------	(0,1) →	(1,0) →	(1,1) →	(2,1) →X	----------	----------
C_{42_P}	-----	----------	--------	----------	(0,5) →	(5,0) →	(5,5) − (0,1) = (5,4) →	(9,5) − (0,2) = (9,3) →
C_{42_11}	----------	----------	----------	----------	(0,2) →	(2,0) →	(2,2) →	(4,2) →X
C_{42_21}	----------	----------	----------	----------	(0,1) →	(1,0) →	(1,1) →X	----------
$C_{42\_31\_P}$	----------	----------	----------	----------	(0,2) →	(2,0) →X	----------	----------
$C_{42\_31\_11}$	----------	----------	----------	----------	(0,1) →	(1,0) →X	----------	----------
C_{32_P}	----------	----------	----------	----------	----------	(0,6) →	(6,0) →	(6,6) − (0,2) = (6,4) →
C_{32_11}	----------	----------	----------	----------	----------	(0,2) →	(2,0) →	(2,2) →X
C_{32_21}	----------	----------	----------	----------	----------	(0,2) →	(2,0) →X	----------
C_{22_P}	----------	----------	----------	----------	----------	----------	(0,10) →	(10,0) →
$C_{22\_11\_P}$	----------	----------	----------	----------	----------	----------	(0,5) →	(5,0) →X
$C_{22\_42\_P}$	----------	----------	----------	----------	----------	----------	(0,4) →	(4,0) →
$C_{22\_42\_11}$	----------	----------	----------	----------	----------	----------	(0,2) →	(2,0) →X

TABLA 1.1 CONTEO DE CONJUNTOS INDEPENDIENTES QUE CORRESPONDE A LA FIGURA 1

LÍNEAS/VÉRTICES	13	23 ↶ 22	33 ↶ 32	43 ↶ 42	44	34 ↶ 33
Lp	(41,29) →	(70,41) − (0,10) = (70,31)	(101,70) − (0,16) = (101,54) →	(155,101) − (0,29) = (155, 72) →	(227,155) →	(382, 227) − (0,54) = (382,173) →
C_{42_P}	(12,9) →	(21,12) − (0,4) = (21,8)	(29,21) →	(50,29) →X	----------	----------
C_{32_P}	(10,6) →	(16,10) →	(26,16) →X	----------	----------	----------
C_{22_P}	(10,10) →	(20,10) →X	----------	---------	---------	----------
$C_{22\_42\_P}$	(4,4) →	(8,4) →X	----------	----------	----------	----------
C_{13_P}	(0,29) →	(29,0) →	(29,29) − (0,6) = (29,23) →	(52,29) − (0,9) = (52,20) →	(72,52) →	(124,72) − (0,23) = (124,49) →
$C_{13\_42\_P}$	(0,9) →	(9,0) →	(9, 9) →	(18, 9) →X	----------	----------
$C_{13\_32\_P}$	(0,6) →	(6,0) →	(6,6) →X	----------	---------	----------
$C_{13\_22\_P}$	(0,10) →	(10,0) →X	----------	----------	---------	----------
$C_{13\_22\_42\_P}$	(0,4) →	(4,0) →X	--------	----------	----------	----------
C_{23_P}	----------	(0,31) →	(31,0) →	(31,31) − (0,8) = (31,23) →	(54,31) →	(85,54) →
$C_{23\_42\_P}$	----------	(0,8) →	(8,0) →	(8,8) →X	----------	-------
$C_{23\_32\_P}$	----------	(0,10) →	(10,0) →X	----------	----------	----------
C_{33_P}			(0,54) →	(54,0) →	(54,54) →	(108,54) →X
$C_{33\_42\_P}$			(0,21) →	(21,0) →X	----------	----------
$C_{33\_13\_P}$			(0,23) →	(23,0) →	(23,23) →	(46,23) →X
$C_{33\_13\_42\_P}$			(0,9) →	(9,0) →X	------	----------
C_{44_p}					(0,155) →	(155,0) →
$C_{44\_13\_P}$					(0,52) →	(52,0) →
$C_{44\_23\_P}$					(0,31) →	(31,0) →
$C_{44\_33\_P}$					(0,54) →	(54,0) →X
$C_{44\_33\_13\_P}$					(0,23) →	(23,0) →X
C_{34_P}						(0, 173) →
$C_{34\_13\_P}$						(0,49) →
$C_{34\_23\_P}$						(0,54) →

TABLA 1.2 CONTEO DE CONJUNTOS INDEPENDIENTES QUE CORRESPONDE A LA FIGURA 1

LÍNEAS/VÉRTICES	24 ↶ 23	14 ↶ 13	15	25 ↶ 24	35 ↶ 34	45 ↶ 44
Lp	(555,382)−(0,85) = (555,297) →	(852,555) − (0,173) = (852, 382) →	(1234,852) →	(2086,1234) − (0,297) = (2086,937) →	(3023,2086) − (0,470) = (3023,1616) →	(4639, 3023) − (0,919) = (4639,2104) →
C_{13_P}	(173,124) →	(297,173) →X	----------	---------	----------	----------
C_{23_P}	(139,85) →X	---------	----------	----------	----------	----------
C_{44_P}	(155,155) − (0,31) = (155,124) →	(279,155) − (0,52) = (279,103) →	(382,279) →	(661,382) − (0,124) = (661,258) →	(919,661) →	(1580,919) →X
$C_{44\_13\_P}$	(52,52) →	(104,52) →X	----------	----------	----------	----------
$C_{44\_23\_P}$	(31,31) →X	----------	----------	----------	----------	---------
C_{34_P}	(173,0) →	(173,173) − (0,49) = (173,124) →	(297,173) →	(470,297) →	(767,470) →X	----------
$C_{34\_13\_P}$	(49,0) →	(49,49) →X	----------	----------	---------	----------
$C_{34\_23\_P}$	(54,0) →X	----------	----------	----------	---------	----------
C_{24_P}	(0,297) →	(297,0) →	(297,297) →	(594,297) →X	----------	----------
$C_{24\_13\_P}$	(0,124) →	(124,0) →X	----------	---------	----------	----------
$C_{24\_44\_P}$	(0,124) →	(124,0) →	(124,124) →	(248,124) →X	----------	-------
$C_{24\_44\_13\_P}$	(0,52) →	(52,0) →X	----------	----------	----------	----------
C_{15_P}	----------		(0,852) →	(852,0) →	(852,852) − (0,173) = (852,679) →	(1531,852) − (0,279) = (1531,573) →
$C_{15\_44\_P}$	----------		(0,279) →	(279,0) →	(279,279) →	(558,279) →X
$C_{15\_34\_P}$	----------		(0,173) →	(173,0) →	(173,173) →X	----------
$C_{15\_24\_P}$	----------		(0,297) →	(297,0) →X	------	----------
$C_{15\_24\_44\_P}$	----------		(0,124) →	(124,0) →X	----------	----------
C_{25_P}	----------			(0,937) →	(937,0) →	(937,937) − (0,258) = (937,679) →
$C_{25\_44\_P}$	----------			(0,258) →	(258,0) →	(516,258) →X
$C_{25\_34\_P}$	----------			(0,297) →	(297,0) →X	---------
C_{35_P}	----------				(0,1616) →	(1616,0) →
$C_{35\_44\_P}$					(0,661) →	(6611,0) →X
$C_{35\_15\_P}$					(0,679) →	(679,0) →
$C_{35\_15\_44\_P}$					(0,279) →	(279,0) →X

TABLA 1.3 CONTEO DE CONJUNTOS INDEPENDIENTES QUE CORRESPONDE A LA FIGURA 1

LÍNEAS/VÉRTICES	46	36 ↶ 35	26 ↶ 25	16 ↶ 15	$i(G)$
Lp	(6743,4639) →	(11382,6743) − (0,1616) = (11382,5127) →	(16509,11382) − (0,2553) = (16509,8829) →	(25338,16509) − (0,5060) = (25338,11449) →	36787
C_{15_P}	(2104,1531) →	(3635,2104) − (0,679) = (3635, 1425) →	(5060,3635) →	(8695,5060) →X	----------
C_{25_P}	(1616,937) →	(2553,1616) →	(4169,2553) →X	----------	----------
C_{35_P}	(1616,1616) →	(3232,1616) →X	----------	----------	----------
$C_{35\_15\_P}$	(679,679) →	(1358,679) →X	----------	----------	----------

5. Conclusiones

En el recorrido sobre los vértices del mallado, se elige visitar en la dirección por columna si $m < n$, en otro caso se puede rotar la malla para tener siempre que se cumple $m \leq n$ y aplicar un recorrido por columna para hacer el cálculo de $i(G_{m,n})$. Para el recorrido presentado, el número máximo de líneas de computación que estarán activas en cualquier momento del proceso de conteo esta superiormente acotado por $O\left(F_{m+3} * (m * n)\right) = O(F_{min\{n,m\}+3} *$

$(m * n)) \cong O((1.1618)^{min\{n,m\}+3} * (m * n))$. Siendo una complejidad muy inferior a la que requiere el método de la matríz de transferencia que es de complejidad exponencial en ambas dimensiones (m y n) de la malla.

Es importante destacar que el tipo de recorrido es vital para obtener complejidades mínimas de tiempo, ya que si se utilizara un recorrido diferente puede incrementarse la complejidad en tiempo de todo el desarrollo. Un trabajo a futuro es dada una malla (inclusive con un número diferente de columnas por renglón), determinar cuál es el recorrido óptimo para obtener complejidades mínimas al realizar el conteo de conjuntos independientes.

Referencias

Arocha J.L., "Propiedades del polinomio independiente de un grafo". Revista Ciencias Matemáticas, Vol. V, 1984, pp. 103-110.

Calkin N., Wilf H. "The number of independent sets in a grid graph". SIAM Journal on Discrete Mathematics, 11, 02 1998.

Dahllöf V., Jonsson P., "An algorithm for counting maximum weighted independent sets and its applications". Proc. Of SODA 2002 thirteenth annual.

Dyer M., Frieze A., Jerrum M., "On counting independent sets in sparse graphs, SIAM J. Comput". Vol.31, No. 5, pp.1527-1541, 2002.

Dyer M., Greenhill C., "Some #P-completeness proofs for colouring and independent sets, research report series, university of Leeds". 1997.

Dyer M., Greenhill C., Corrigendum: "The complexity of counting graph homomorphism, RSA: Random Strutures and Algorithms", pp.346-352, 2004.

Greenhill Catherine. "The complexity of counting colouring and independent sets in sparse graphs and hypergraphs" Computational Complexity, pp.52-72, 2000.

Guillen C., López A., and G. De Ita. "Computing #2-sat of grids, grid-cylinders and grid-tori boolean formulas". In Proc. of the 15th RCRA workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion, Vol. 451. CEUR-WS.org, 2008.

Luby M., Vigoda E. "Approximately counting up to four, Twenty-ninth annual Symp. On theory of computing". ACM NEW York, 1997, pp.682-687.

Mordecai Golin, Yiu Cho Leung, Yajun Wang, and Xuerong Yong. "Counting structures in grid graphs, cylinders and tori using transfer matrices: Survey and new result", In ALENEX/ANALCO, pages 250–258, 01 2005.

OkamotoY., Uno T., Uehara R., "Counting the number of independent sets in chordal graphs". Proc. of the 31st Int. WS on Graph-Theoretic concepts in computer science. WG 2005, pp. 433-444, Lecture notes in computer science Vol. 3787, 2005.

Reinhardt Euler. "The Fibonacci number of a grid graph and a new class of integer sequences". Journal of Integer Sequences, 8, 05 2005.

Roth D., "On the hardness of approximate reasoning". Artificial intelligence 82, 1996, pp.273-302.

Russ B., “Randomized Algorithms: Approximation, generation, and counting, distinguished dissertations”. Springer, 2001.
R.M. Roth, P.H. Siegel, and J.K. Wolf. Efficient coding schemes for the hard-square model, IEEE Trans. Inform. Theory, 47:1166-1176, 2001.
Vadhan Salil P., The complexity of counting in sparse, regular, and planar graphs, SIAM Journal on Computing, Vol. 31, No.2, pp. 398.427, 2001
Zacarias F., Moyao Y., Contreras M., De Ita G., Bello P., “Combinational algorithms and learning”, Montiel & Soriano, 2017, pp 33-44.
Zuhe Zhang, "Merrifield-Simmons index and its entropy of the 4-8-8 lattice", J Statical Physics 154, (2014) 11131123.

Capítulo 9

Reconociendo topologías extremas con respecto al índice Merrifield-Simmons en grafos bipoligonales

Guillermo De Ita, Meliza Contreras, Pedro Bello

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

deitaluna63@gmail.com, vikax68@gmail.com, pb5pbello@gmail.com

Resumen. Mostramos cómo las propiedades de la secuencia $\beta_{i,j}$ que representa el producto entre dos números Fibonacci ($F_i \cdot F_j$) se puede utilizar para el cálculo del índice de Merrifield-Simmons en grafos bipoligonales. Nuestro método no requiere el cálculo explícito del número de conjuntos independientes de los grafos involucrados, sino que se basa en la aplicación de la regla de división de aristas como una forma de descomponer el grafo inicial. Mostramos que los valores extremos para los grafos bipoligonales se encuentran en dos columnas consecutivas; el valor extremo mínimo en $\beta_{3,k-3}$ y el valor extremo máximo en $\beta_{4,k-4}$.

Palabras Clave: Conteo de conjuntos independientes, índice merrifield-Simmons, topologías extremales, grafos bipoligonales.

1 Introducción

El reconocimiento de topologías extremas en grafos ha representado un estudio significativo en el área de reconocimiento de patrones estructurales (Wagner y Gutman, 2010). Especialmente, en la teoría de grafos, diversos trabajos se ocupan de la caracterización de grafos extrémales con respecto a los índices de Hosoya y Merrifield-Simmons para diferentes topologías de grafos, como son los árboles, grafos unicíclicos y ciertas estructuras que contienen ciclos pentagonales y hexagonales.

Merrifield y Simmons mostraron la correlación entre el número de conjuntos independientes de G, denotados $i(G)$, y los puntos de ebullición del grafo molecular representado por G (Merrifield y Simmons, 1989). Esta es una de las principales razones por las que al número de conjuntos independientes de un grafo G, en el área de la química matemática, se le llama el índice de Merrifield-Simmons (M-S) de G. Sin embargo, en el área de la teoría de grafos, a $i(G)$ se le llama el número Fibonacci de G.

Los números Fibonacci y sus propiedades han sido útiles en el análisis de compuestos estructurales en el área de la química matemática. El índice de Merrifield-Simmons es un

índice topológico significativo de la química estructural del grafo molecular *G* (Deng, 2009; Wagner y Gutman, 2010). Un índice topológico es un mapeo del conjunto de compuestos químicos representados por grafos moleculares al conjunto de números reales. Muchos índices topológicos están estrechamente correlacionados con algunas características fisicoquímicas de los compuestos subyacentes. El índice M-S y el índice Hosoya son algunos de los índices topológicos más populares en química (Li et al, 2005).

Los arreglos de grafos hexagonales han sido ampliamente investigados, y representan un área relevante de interés en la química matemática porque se han utilizado para estudiar las propiedades intrínsecas de los sistemas de Bencenos. Por ejemplo, un fenileno es cualquiera de los aromáticos radicales divalentes que se obtienen a partir de una molécula de benceno mediante la eliminación de dos átomos de hidrógeno. Muchos de los polímeros en los que el bloque de construcción básico es un fenileno se llama polifenileno. Los polifenilenos constituyen una clase importante de compuestos que sirven como precursores de muchos materiales científicos y comercialmente interesantes, como es el caso del óxido de polifenileno y el sulfuro de polifenileno.

La clase especial de grafos representados por dos polígonos unidos por una arista son los grafos básicos que representan compuestos de polifenilenos. Los polifenilenos no ramificados aparecen en el contexto de conductores orgánicos de baja dimensión, mientras que sus contrapartes similares a los dendrímeros, desempeñan un papel importante en la síntesis de grandes moléculas de grafeno (Döslić y Litz, 2012).

Varios trabajos tratan sobre la caracterización de grafos extremos con reminiscencias a los índices de Hosoya y Merrifield-Simmons en clases de grafos [(Yuefen y Fuji, 2017); (Deng, 2010); (Döslić y Litz, 2012); (De_Ita, 2017); (De_Ita, 2018); (Wagner y Gutman, 2010)]. Por ejemplo, en (Ren y Zhang, 2007) se determinó el índice mínimo de Merrifield-Simmons de cadenas hexagonales dobles. Gutman et al, 2005 caracterizó el árbol con el índice máximo de Merrifield-Simmons entre los árboles con un diámetro dado. En (Zhu et al, 2010) se realiza una encuesta sobre grafos extremos para índices de Hosoya y Merrifield-Simmons que involucran diferentes topologías de grafos.

Por otro lado, hay varios trabajos que analizan secuencias de productos entre los números de Fibonacci. Por ejemplo, (Adegoke, 2017) derivó identidades de productos infinitos que involucran números de Fibonacci y de Lucas; en (Bulawa y Lee, 2017), se muestra que la función generadora de la secuencia de Fibonacci produce valores que constituyen todos los números racionales; en (Edgar, 2016) se desarrolla una generalización sobre dos identidades probadas de Fibonacci-Lucas. En (Melham, 2016), el resultado principal es la obtención de una identidad para la *m*-ésima potencia de los números de Fibonacci en la que los subíndices de los números de Fibonacci involucrados están espaciados arbitrariamente, así como sus identidades duales.

Con respecto a los valores máximos y mínimos de Fibonacci, en (Berenhaut et al, 2011) se demostraron varios resultados sobre la convergencia del mínimo y el máximo de recurrencias de orden superior, mientras que en (Ando, 1995) algunas propiedades del sistema de secuencias *m* se definieron a través de una relación de recurrencia que utiliza una relación de producto matricial.

En este artículo, mostramos cómo las propiedades sobre el producto entre dos números de Fibonacci y la aplicación de la regla de división de aristas se pueden utilizar para el cálculo de valores extremos del índice de Merrifield-Simmons en un fenileno. En particular, reconocemos los casos extremos para un grafo bipoligonal.

2 Preliminares

Sea $G = (V, E)$ un grafo no dirigido con conjunto de vértices V y conjunto de aristas E. Se supone que G es un grafo simple que no tiene bucles ni aristas paralelas. El vecindario de $x \in V$ es el conjunto $N(x) = \{y \in V: xy \in E\}$, y su vecindad cerrada es $N(x) \cup \{x\}$ el cuál se denota por $N[x]$. El grado de un vértice x en un grafo G, que se denota por $d_G(x)$, es $|N(x)|$. Cuando no hay duda a que subgrafo G nos referimos, entonces se omite el subíndice de G. El grado del grafo G es $\Delta(G) = \max\{d(x): x \in V\}$.

Un camino entre dos vértices v y w, denotado como P_{vw} o simplemente como P_n, es una secuencia de las aristas: v_0v_1, v_1v_2, . . . , $v_{n-1}v_n$ tal que $v = v_0$, $v_n = w$ y $v_kv_{k+1} \in E$ para $0 \leq k < n$; la longitud del camino es el número de aristas en el camino P_n. Un camino simple es un camino donde $v_0, v_1, \ldots, v_{n-1}, v_n$ son todos distintos. Un ciclo es un camino no vacío tal que el primer y el último vértice son idénticos, y un ciclo simple es un ciclo en el que no se repite ningún vértice, con la excepción de que el primer y el último vértice son idénticos.

Un subconjunto $S \subseteq V$ se llama independiente si para cada $u, v \in S$ se implica que $uv \notin E$. El problema correspondiente al conteo de conjuntos independientes, denotado por $i(G)$, consiste en contar el número de conjuntos independientes de un grafo G. Calcular $i(G)$ es un problema #P-completo para grafos G donde $\Delta(G) \geq 3$. Calcular $i(G)$ permanece #P-completo incluso si está restringido a grafos 3-regulares (Dyer y Greenhill, 1997).

Sea $G = (V, E)$ un grafo molecular. Se denota por $n(G, k)$ el número de formas en que k vértices mutuamente independientes se pueden seleccionar en G. Por definición, $n(G, 0) = 1$ para todo los grafos y $n(G,1)=|V(G)|$. Además, $i(G) = \Sigma_{k\geq 0}\, n(G, k)$ es el índice Merrifield-Simmons de G, esto es exactamente el número de conjuntos independientes de G.

Un polígono (también llamado grafo poligonal) es un ciclo simple. Por lo tanto, un grafo de ciclo C_n de longitud n representa un polígono de n lados, y forma un n-ágono. La forma en que se unen dos k-ágonos, a través de un vértice común o a través de una arista común, describe a diferentes compuestos químicos. Dos polígonos que tienen una arista en común se llaman adyacentes.

Una cadena poligonal es un grafo simple G 2-conectado que se obtiene identificando un número finito de polígonos regulares congruentes (llamados polígonos básicos) uno por uno tal que cada vértice de G tiene grado 2 o 3 y cada polígono básico, excepto el primero y el último, es adyacente a exactamente dos polígonos básicos. Un arreglo poligonal es un grafo $P_{k,t}$ obtenido mediante la identificación de un número finito de polígonos congruentes t. Cuando cada polígono en $P_{k,t}$ tiene el mismo número de lados k, entonces $P_{k,t}$ se convierte en una cadena de t k-ágonos.

Algunas reglas de reducción han sido útiles para contar objetos combinatorios en grafos. En particular, las siguientes reglas son utilizadas comúnmente:

1. Regla de reducción de vértices: sea $v \in V(G)$, $i(G) = i(G - v) + i(G - (N[v]))$.
2. Regla de división de aristas: sea $e = \{x, y\} \in E(G)$, $i(G) = i(G - e) - i(G - (N[x] \cup N[y]))$.

3 El producto entre números de Fibonacci con índices complementarios

Denotemos el n-ésimo número de Fibonacci como F_n. La serie de Fibonacci se define como: $F_0 = 0$; $F_1 = 1$ y $F_n = F_{n-1} + F_{n-2}$. La fuerte relación entre el número de conjuntos independientes de un grafo $i(G)$ y los números de Fibonacci es ampliamente conocida. Por ejemplo, $i(P_n) = F_{n+2}$, $i(C_n) = F_{n+1} + F_{n-1}$, donde P_n y C_n son el camino y el ciclo de n vértices, respectivamente. Consideremos un vértice aislado como un camino lineal de longitud cero, por lo tanto, $i(P_1) = F_3 = 2$.

En [(De Ita, 2017), (De Ita, 2018)], se muestran algunas propiedades sobre la secuencia $\beta_{s,k} = F_k * F_{k-s}$, para $k > 0$, $1 \leq s \leq k - 1$. Por ejemplo, el comportamiento simétrico de la secuencia $\beta_{s,k}$ en la posición $s > \lfloor k/2 \rfloor$. Por tanto, $\beta_{k,\lfloor k/2 \rfloor - j} = \beta_{k,\lfloor k/2 \rfloor + j}$ si k es par y $\beta_{k,\lfloor k/2 \rfloor - j} = \beta_{k,\lfloor k/2 \rfloor + j + 1}$ sí k es impar y para j tal que $1 \leq j \leq \lfloor k/2 \rfloor - 2$.

Además, la secuencia $\beta_{k,s}$ es creciente para los índices pares de s y tiene un comportamiento decreciente para los índices impares de s. Por ejemplo $\beta_{k,2p} < \beta_{k,2p+1}$ para $p \in \{0, 1, \ldots, \lfloor k/4 \rfloor\}$, y para toda k. Mientras que $\beta_{k,2p+1} > \beta_{t,2p+3}$ para $p \in \{0, 1, \ldots, \lfloor k/4 \rfloor - 1\}$ y para toda k.

En la Tabla 1, presentamos algunos de los valores de la secuencia $\beta_{s,k}$. Note que las diferentes relaciones pueden ser inferidas cuando consideramos los valores de los rangos como son utilizados con el triángulo de Pascal.

n	F_n	$\beta_{1,k}$	$\beta_{2,k}$	$\beta_{3,k}$	$\beta_{4,k}$	$\beta_{5,k}$	$\beta_{6,k}$	$\beta_{7,k}$	$\beta_{8,k}$	$\beta_{9,k}$	$\beta_{10,k}$	$\beta_{11,k}$	$\beta_{12,k}$
		Max	*Min*										
1	1	0											
2	1	1	0										
3	2	1	1	0									
4	3	2	1	2	0								
5	5	3	2	2	3	0							
6	8	5	3	4	3	5	0						
7	1 3	8	5	6	6	5	8	0					
8	2 1	13	8	10	9	10	8	13	0				
9	3 4	21	13	16	15	15	16	13	21	0			
1 0	5 5	34	21	26	24	25	24	26	21	34	0		
1 1	8 9	55	34	42	39	40	40	39	42	34	55	0	
1 2	144	89	55	68	63	65	64	65	63	68	55	89	0

Tabla 1. El producto de dos Fibonacci con índices complementarios

Otros resultados relevantes (De Ita, 2018) es que $\beta_{1,k} = F_1 \cdot F_{k-1} = F_{k-1}$ es máximo para la serie, mientras que $\beta_{2,k} = F_2 \cdot F_{k-2} = F_{k-2}$ es mínimo para la misma serie en la fila (k).

Note que los siguientes valores extrémales en $\beta_{s,k}$ corresponde con los valores máximos para $\beta_{3,k} = F_3 \cdot F_{k-3} = 2 \cdot F_{k-3}$ y con los valores mínimos para y $\beta_{4,k} = F_4 \cdot F_{k-4} = 3 \cdot F_{k-4}$, manteniendo la misma fila (k).

En particular, el máximo $F_1 \cdot F_{k-1} = F_{k-1}$ para la fila (k) de la tabla resulta ser el mínimo $F_2 \cdot F_{k-1} = F_{k-1}$ para la fila ($k + 1$). Además, la diferencia entre el máximo y el mínimo en la fila (k) es $F_{k-1} - F_{k-2} = F_{k-3}$. El hecho de que los valores extremos de $\beta_{k,s}$ estén en las dos primeras columnas consecutivas de la Tabla 1 y los siguientes valores extremos también están en las siguientes dos columnas siguientes, tendrá consecuencias lógicas en las topologías que representan los valores extremos para el índice de Merrifield-Simmons en grafos bipoligonales.

Estos resultados se consideran aquí para mostrar nuevas propiedades para $\beta_{s,k}$ que serán útiles en nuestro análisis. Por ejemplo, en la siguiente sección mostramos cómo aplicar algunas de las propiedades de la serie $\beta_{s,k}$ para determinar topologías extremas sobre grafos bipoligonales.

4 Grafos bipoligonales

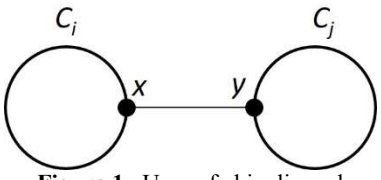


Figura 1. Un grafo bipoligonal

Sea C_i y C_j dos polígonos con el número respectivo de vértices i y j. Se forma una clase especial de grafos para unir C_i y C_j a través de una arista $e = \{x, y\}$, con $x \in V(C_i)$ e $y \in V(C_j)$, ver Figura 1. Llamamos a esta clase de grafos conectados a través de una arista de corte, un grafo bipoligonal, y se denotará por $H_{i,j}$. Especialmente, cuando los polígonos C_i y C_j son hexágonos, $H_{i,j}$ es el grafo primitivo utilizado para formar cadenas de fenilenos y bypiridinas.

Consideremos ahora la regla de división de arista: sea $e = \{x, y\} \in E(G)$, entonces $i(G)=i(G\text{-}e)\text{-}i(G\text{-}(N[x] \cup N[y]))$. Mostramos la aplicación de la regla de división de arista con el fin de contar el índice de Merrifield-Simmon en fenilenos.

Proposición 1. $i(H_{i,j}) = F_{i+1} \cdot F_{j+1} + F_{i+1} \cdot F_{j-1} + F_{i-1} \cdot F_{j+1}$

Prueba. De acuerdo con la regla de división de arista (consulte la Figura 2):

$i(H_{i,j}) = i(C_i) \cdot i(C_j) - i(P_{i-3} \cdot P_{j-3}) = (F_{i+1} + F_{i-1}) \cdot (F_{j+1} + F_{j-1}) - F_{i-1} \cdot F_{j-1}$

$= F_{i+1}F_{j+1} + F_{i+1}F_{j-1} + F_{i-1}F_{j+1} + F_{i-1}F_{j-1} - F_{i-1}F_{j-1} = F_{i+1}F_{j+1} + F_{i+1}F_{j-1} + F_{i-1}F_{j+1}$.

Este resultado se puede ver como $F_{i+1}\cdot\ (F_{j+1}+F_{j-1}) + F_{i-1}\cdot\ F_{j+1}$ donde $(F_{j+1} + F_{j-1})$ representa el $j-$ésimo número de Lucas que denotaremos por L_j.

Sea k = i + j, fijando $k \geq 6$, consideramos los diferentes subgrafos formados por las variaciones: $3 \leq i, j \leq (k - 3)$. Consideramos que $H_{i,j}$ se puede descomponer en todas las combinaciones posibles de polígonos C_i y C_j, manteniendo $i + j$ como una constante.

En la Tabla 2 mostramos todas las combinaciones posibles para conformar C_i y C_j y fijando como constante el número total de vértices en C_i y C_j (en este caso, 12 vértices). En la Tabla 2 también mostramos cómo calcular su número respectivo de conjuntos independientes basados en la proposición anterior.

En las siguientes proposiciones se demuestra qué combinación de polígonos proporciona los conjuntos independientes máximos y mínimos para cualquier valor de k (número total de vértices).

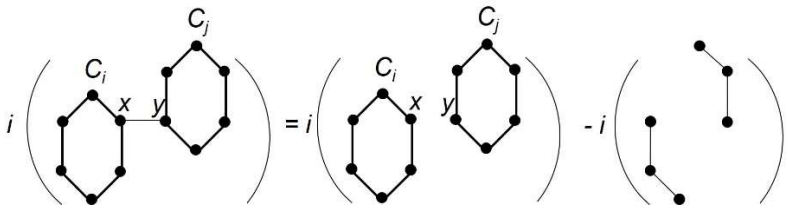


Figura 2. Aplicando la regla de división por arista a un grafo bipoligonal

Teorema 1. El mínimo $i(H_{i,j}) = \min\{i(H_{r,s}) :\ r + s = k,\ r, s \geq\ 3\} = i(H_{3,k-3})$

Prueba. *Sea* $i + j = k$, $i = 3$, $j = k - 3$, $i(H_{3,k-3}) = F_{3+1}\cdot\ F_{k-3+1} + F_{3+1}\cdot\ F_{k-3-1} + F_{3-1}\cdot F_{k-3+1}$, *debido a la proposición 3.1. Por lo tanto,* $i(H_{3,k-3}) = F_4\cdot\ F_{k-2} + F_4\cdot\ F_{k-4} + F_2\cdot F_{k-2} = F_4\cdot\ (F_{k-2} + F_{k-4}) + F_2\cdot\ F_{k-2}$.

Si asumimos que $i > 3$, *entonces* $i(H_{i,k-i}) = F_{i+1}(F_{k-i+1} + F_{k-i-1}) + F_{i-1}\cdot\ F_{k-i+1}$. *Tenemos que,* $F_2\cdot F_{k-2} < F_{i-1}\cdot F_{k-i+1},\ \forall i > 3$, *dado que* $F_2\cdot F_{k-2}$ *es el mínimo de la serie* $\beta_{s,k}$.

Por otro lado, $F_4\cdot\ (F_{k-2} + F_{k-4}) = F_4\cdot\ L_{k-3}$ *y* $F_{i+1}\cdot\ (F_{k-i+1} + F_{k-i-1}) = F_{i+1}\cdot L_{k-i}$, *considerando que* $i \geq 3$. *Así* $F_4\cdot L_{k-3} < F_{i+1}\cdot L_{k-i},\ \forall i > 3$ *porque* $\beta_{s,k}$ *es creciente en los índices pares de s y entonces* $F_4\cdot\ (F_{k-2} + F_{k-4})$ *es el siguiente valor mínimo para cualquier par* $F_{i+1}\cdot\ (F_{k-i+1} + F_{k-i-1})$, *con* $i \geq 3$. *Sin considerar el mínimo valor de* $F_2\cdot\ (F_{k-2} + F_{k-4})$ *en la fila (k) porque este valor no representa ninguna descomposición bipoligonal.*

Teorema 2. El máximo $i(H_{i,j}) = \max\{i(H_{r,s}) :\ r + s = k,\ r, s \geq\ 4\} = i(H_{4,k-4})$

Prueba. *Sea* $i + j = k$, $i = 4$, $j = k - 4$. *Debido a la proposición 3.1,*

$i(H_{4,k-4}) = F_{4+1}\cdot\ F_{k-4+1} + F_{4+1}\cdot\ F_{k-4-1} + F_{4-1}\cdot\ F_{k-4+1} = F_5\cdot\ F_{k-3} + F_5\cdot\ F_{k-5} + F_3\cdot F_{k-3} = F_5\cdot (F_{k-3} + F_{k-5}) + F_3\cdot\ F_{k-3}$. *Si asumimos que* $i > 4$ *entonces* $i(H_{i,k-i}) = F_{i+1}(F_{k-i+1} + F_{k-i-1}) + F_{i-1}\cdot\ F_{k-i+1}$, *debido a la proposición 3.1. Tenemos que,* $F_3\cdot F_{k-3} > F_{i-1}\cdot F_{k-i+1}$, *desde* $F_3\cdot F_{k-3}$ *es el siguiente valor máximo en la serie* $\beta_{s,k}$ *: dado que el valor* $F_1\cdot F_{k-1}$ *no representa ninguna descomposición bipoligonal.*

Por otro lado, $F_5 \cdot (F_{k-3} + F_{k-5}) = F_5 \cdot L_{k-4}$ *y* $F_{i+1} \cdot (F_{k-i+1} + F_{k-i-1}) = F_{i+1} \cdot L_{k-i}$, *considerando* $i \geq 4$. *Así,* $F_5 \cdot L_{k-4} > F_{i+1} \cdot L_{k-i}$, $\forall i > 4$, *porque* $\beta_{s,k}$ *es decreciente sobre los índices impares de s. Por lo tanto,* $i(H_{4,k-4}) = F_5 \cdot (F_{k-3} + F_{k-5}) + F_3 \cdot F_9$ *es el siguiente valor máximo, sin considerar* $F_3 \cdot (F_{k-3} + F_{k-5})$ *para cualquier par* $F_{i+1} \cdot (F_{k-i+1} + F_{k-i-1})$, *con* $i \geq 4$.

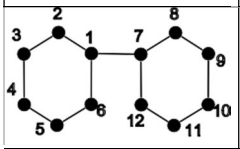
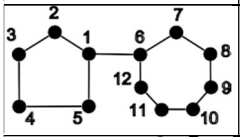
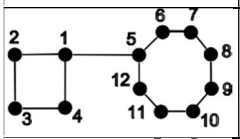
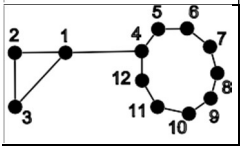
***Topología* poligonal**	i	j	$F_{i+1}F_{j+1} + F_{i+1}F_{j-1} + F_{i-1}F_{j+1}$	$i(G)$
	6	6	$F_7F_7 + F_7F_5 + F_5F_7$	169 + 65 + 65 = 299
	5	7	$F_6F_8 + F_6F_6 + F_4F_8$	168 + 64 + 63 = 295
	4	8	$F_5F_9 + F_5F_7 + F_3F_9$	170 + 65 + 68 = 303
	3	9	$F_4F_{10} + F_4F_8 + F_2F_{10}$	165 + 63 + 55 = 283

Tabla 2. Combinaciones de grafos bipoligonales con el mismo número de vértices totales.

En la Tabla 2, consideramos que el número de vértices k para el grafo bipoligonal es 12. Mostramos que no es necesario hacer el cálculo explícito de $i(G)$, sino que necesitamos conocer los valores de la secuencia $\beta_{s,k}$. Para la instancia analizada, los valores extremos se identifican cuando se obtiene la mayor variación (entropía) entre los tamaños de los Polígonos. En este caso, el mínimo valor corresponde a $|C_j| - |C_i| = 6$ y el máximo valor es cuando $|C_j| - |C_i| = 4$.

Cuando $k = 12$ hay otras instancias diferentes de cadenas poligonales, por ejemplo, una cadena de 3 cuadrados o una cadena de 4 triángulos. Para esos casos, se obtienen diferentes valores para $i(G)$. Para la cadena de 3 cuadrados tenemos que $i(G) = 287$ y para una cadena de 4 triángulos $i(G) = 209$. Pero en nuestro estudio, queremos mantener la estructura de grafos bipoligonales.

Para grafos bipoligonales, podemos construir una serie de valores $H_{i,j}$ para el índice Merrifield-Simmon del grafo bipoligonal, donde el primer polígono tiene i vértices y el segundo j vértices. En este caso, la serie tiene un comportamiento similar al de la serie $\beta_{i,j}$

de la Tabla 1. En la Tabla 3, mostramos los valores de la serie $i(H_{i,j})$ con sus valores ordenados como en la Tabla 1, en forma del triángulo sobre los valores de $i(H_{i,j})$.

k	***Min***	***Max***						
8	$i(H_{3,5})$	$i(H_{4,4})$	$i(H_{5,3})$					
9	$i(H_{3,6})$	$i(H_{4,5})$	$i(H_{5,4})$	$i(H_{6,3})$				
10	$i(H_{3,7})$	$i(H_{4,6})$	$i(H_{5,5})$	$i(H_{6,4})$	$i(H_{7,3})$			
11	$i(H_{3,8})$	$i(H_{4,7})$	$i(H_{5,6})$	$i(H_{6,5})$	$i(H_{7,4})$	$i(H_{8,3})$		
12	$i(H_{3,9})$	$i(H_{4,8})$	$i(H_{5,7})$	$i(H_{6,6})$	$i(H_{7,5})$	$i(H_{8,4})$	$i(H_{9,3})$	
13	$i(H_{3,10})$	$i(H_{4,9})$	$i(H_{5,8})$	$i(H_{6,7})$	$i(H_{7,6})$	$i(H_{8,5})$	$i(H_{9,4})$	$i(H_{10,3})$

Tabla 3. El índice de Merrifield-Simon para grafos bipoligonales con un número total de k vértices

La Tabla 3 tiene las mismas propiedades que se mostraron en la Tabla 1. En este caso, $i(H_{i,j})$ será una serie creciente sobre los índices impares en i, y una serie decreciente sobre los índices pares de j. Los valores extremos en esta serie se encuentran en las dos primeras columnas; el extremo mínimo en $i(H_{3,k-3})$ y el extremo máximo en $i(H_{4,k-4})$.

5 Conclusiones

Hemos mostrado cómo las propiedades de la secuencia $\beta_{s,k}$, que representa el producto entre los números de Fibonacci: F_s y F_{k-s}, se pueden utilizar para el cálculo del índice de Merrifield-Simmons en grafos bipoligonales. Nuestro método no requiere el cálculo explícito del número de conjuntos independientes de los grafos involucrados, sino que se basa en la aplicación de la regla de división de arista como una forma de descomponer el grafo inicial. Mostramos que los valores extremos para grafos bipoligonales se encuentran en las primeras dos columnas consecutivas; el valor extremo mínimo en $\beta_{3,k-3}$ y el valor extremo máximo en $\beta_{4,k-4}$.

Referencias

Adegoke, K. (2017). "Some Infinite Product Identities Involving Fibonacci and Lucas Numbers", *Fibonacci Quaterly*, vol. 55(4), pp. 343-351.

Ando, S. (1995). "On a system of sequences defined by a recurrence relation", *Fibonacci Quaterly*, vol 33(3), pp. 279-282.

Berenhaut, K.S., Magargeei, M. y Rabidous, S. M. (2011). "Asymptotic behavior of solutions to minimum-maximum recurrences of higher-order", *Aportaciones Matemáticas Investigación*, vol 20, pp. 45-51.

Bulawa, A. y Lee, W. K. (2017). "Integer Values of Generating Functions for the Fibonacci and Related Sequences", *Fibonacci Quaterly*, vol. 55(1), pp. 74-81.

De_Ita, G., Marcial, J. R, Bello, P. y Contreras, M. (2018). "Linear-time Algorithms for Computing the Merrifield-Simmons Index on Polygonal Trees", *MATCH Commun. Math. Comput. Chem.*, vol. 79, pp. 55-78.

De_Ita, G., Marcial, J. R., Hernández, J. A., Valdovino, R. M. y Romero, M. (2017). "Extending Extremal Polygonal Arrays for the Merrifield-Simmons Index", *Lecture Notes in Computer Science*, vol. 10267, pp. 22-31.

Deng, H. (2009). "The smallest Merrifield-Simmons index of (n; n+1)-graphs", *Mathematical and Comp. Modelling*, vol. 49 (1,2), pp. 320-326.

Deng, H. (2010). "Catacondensed Benzenoids and Phenylenes with the Extremal Thirdorder Randic Index1", *Comm. in Math. and in Comp. Chem*., vol. 64, pp. 471-496.

Döslić, T. y Litz, M. S. (2012). "Matchings and Independent Sets in Polyphenylene Chains", *MATCH Commun. Math. Comput. Chem*, vol. 67, pp. 313-330.

Dyer M. y Greenhill C. (1997). "Some #P-completeness Proofs for Colourings and Independent Sets", *Research Report Series*, University of Leeds, 1997.

Edgar, T. (2016). "Extending Some Fibonacci-Lucas Relations", *Fibonacci Quaterly*, vol. 54(1), pp. 79.

Gutman, I., Li, X. y Zhao, H. (2005). "On the Merrifield-Simmons index of trees", *MATCH Commun. Math. Comput. Chem*. vol. 54, pp. 389-402.

Li, X., Zhao, H. y Gutman I. (2005). "On the Merrifield-Simmons index of trees", *MATCH Commun, Math. Comput. Chem*, vol. 54, pp. 389-402.

Melham, R. S. (2016). "New Identities Satisfied by Powers of Fibonacci and Lucas Numbers", *Fibonacci Quaterly*, vol. 54(4), pp. 296-303.

Merrifield, R. E. y Simmons, H. E. (1989). "Topological Methods in Chemistry", *Wiley, New York*.

Ren, H. y Zhang, F. (2007). "Double hexagonal chains with maximal Hosoya index and minimal Merrifield-Simmons index", *J. Math. Chem.* vol. 42(4), pp. 679-690.

Wagner, S. y Gutman, I. (2010). "Maxima and minima of the Hosoya index and the Merrifield-Simmons index", *Acta Appl. Math*. Vol. 112(3), pp. 323-346.

Yuefen, C. y Fuji, Z. (2017). "Extremal polygonal chains on k-matchings", *MATCH Commun. Math. Comput. Chem*, vol. 79, pp. 55-78.

Zhu, Z., Li, S. y Tan, L. (2010). "Tricyclic graphs with maximum Merrifield-Simmons index", *Discrete Applied Mathematics*, vol. 158, pp. 204-212.

Índice de autores

Nombre del Autor	Nacionalidad	
Ana Laura Lezama Sánchez	Mexicana	
Beatríz Bernabé Loranca	Mexicana	
Darnes Vilariño Ayala	Mexicana	
Domingo Rodríguez Benavides	Mexicana	
Erick Barrios González	Mexicana	
Fernando Zacarias Flores	Mexicana	Editor
Gabriela A. García Robledo	Mexicana	
Guillermo De Ita Luna	Mexicana	Editor
José Alejandro Reyes Ortiz	Mexicana	
Josué Padilla Cuevas	Mexicana	
Luis Carlos Altamirano Robles	Mexicana	Editor
Luis Fernando Hoyos Reyes	Mexicana	
Luis Filiberto Regino Medina	Mexicana	
María Auxilio Medina Nieto	Mexicana	
María Josefa Somodevilla García	Mexicana	
Maricela Bravo	Mexicana	
Meliza Contreras Gónzalez	Mexicana	Editora
Mireya Tovar Vidal	Mexicana	Editora
Pedro Bello López	Mexicana	Editor
Pierre Antoine Delice	Haitiano	
Yolanda Moyao Martínez	Mexicana	Editora

Compiladores

Mireya Tovar Vidal
Guillermo De Ita Luna
Pedro Bello López
Meliza Contreras Gónzalez
Fernando Zacarias Flores
Yolanda Moyao Martínez
Luis Carlos Altamirano Robles

Revisores

Beatriz González Beltrán
Carmen Cerón Garnica
Claudia Zepeda Cortés
Fernando Zacarías Flores
Gabriela Alejandra García Robledo
Guillermo De Ita Luna
Hilda Castillo Zacatelco
Hugo Pablo Leyva
Jacobo Leonardo González Ruiz
José Alejandro Reyes Ortiz
José Antonio Hernández Servín
José David Alanís Urquieta
José de Jesús Lavalle Martínez
José Luis Carballido Carranza
José Raymundo Marcial Romero
Josue Padilla Cuevas
Karina Rosales López
Leonardo Sánchez Martínez
María Auxilio Medina Nieto
María Beatríz Bernábe Loranca
Maricela Bravo
Mario Rossainz López
Meliza Contreras González
Mireya Tovar Vidal
Oscar Herrera Alcantara
Pedro Bello López
Rogelio González Velázquez
Rosa María Valdovinos Rosas
Yolanda Moyao Martínez

Editores

Mireya Tovar Vidal
Guillermo De Ita Luna
Pedro Bello López
Meliza Contreras Gónzalez
Fernando Zacarias Flores
Yolanda Moyao Martínez
Luis Carlos Altamirano Robles

Procesamiento de lenguaje natural y métodos basados en grafos
Coordinado por
Mireya Tovar Vidal
Guillermo De Ita Luna
Pedro Bello López
Meliza Contreras Gónzalez
Fernando Zacarias Flores
Yolanda Moyao Martínez
Luis Carlos Altamirano Robles
está disponible en PDF en la página
de la Facultad de Ciencias de la Computación
de la Benemérita Universidad Autónoma de Puebla (BUAP)
https://www.cs.buap.mx/~mtovar/doc/Libros/LibroICOKG22.pdf
a partir de diciembre de 2022
Peso del archivo: 8.5 MB