UCAB Universidad Católica ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA INFORMÁTICA

Algoritmo de minería de datos bibliográficos para sugerir un conjunto ordenado de artículos científicos acerca de un tema de investigación predefinido

Este Jurado; una vez realizado el examen del presente trabajo ha evaluado su contenido con el resultado: Diecinueve (19) Mención Honorífica

J U R A D O E X A M I N A D O R

Firma:	Firma:	Firma:
Nombre: Alfredo Benso (RAFAEL LARA CAMPOS)	Nombre: Wilmer Pereira	Nombre: Henry Martínez

REALIZADO POR	**MARTÍNEZ, Alejandro**
PROFESOR GUÍA	**BENSO, Alfredo**
FECHA	**Febrero, 2017**

UCAB Universidad Católica
ANDRÉS BELLO
Facultad de Ingeniería
Escuela de Ingeniería Informática

Algoritmo de minería de datos bibliográficos para sugerir un conjunto ordenado de artículos científicos acerca de un tema de investigación predefinido

TRABAJO ESPECIAL DE GRADO

presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

como parte de los requisitos para optar al título de

INGENIERO EN INFORMÁTICA

REALIZADO POR: MARTÍNEZ, Alejandro

TUTOR ACADEMICO BENSO, Alfredo

FECHA Febrero, 2017.

AGRADECIMIENTOS

A mi mamá y a mi papá, por darme todo para elegir mí destino.

A Maricarmen Lecuna, por haberme motivado a iniciar esta maravillosa experiencia, y por haberme acompañado.

A cada uno de mis amigos, por su amistad y apoyo durante la tesis.

A mi tutor, por la inspiración, orientación, la oportunidad, y por la confianza depositada en mí

A todos y cada uno de mis profesores, por haber contribuido a mi formación. En especial a mi querida profe Susana García, por sus ganas y apoyo desde el anteproyecto.

Finalmente, a Gema Páez, por ayudarme a encontrar mi fuerza interior para terminar la tesis.

Lo mejor está por venir, y es en gran parte por su positiva contribución a mí.

Alejandro Martínez

INDICE DE CONTENIDO

AGRADECIMIENTOS i
INDICE DE CONTENIDO ii
ÍNDICE DE TABLAS v
ÍNDICE DE FIGURAS vi
SIPNOSIS vii
INTRODUCCIÓN 1
CAPÍTULO I. Problema 2
I.1. Planteamiento del Problema 2
I.2. Objetivos 3
I.2.1. Objetivo General 3
I.2.2. Objetivos Específicos 3
I.3. Alcance 4
I.4. Limitaciones 6
I.5. Justificación 7
CAPÍTULO II. Marco Teórico 8
II.1 Minería de Datos 8
II.2 Grafos 8
II.3 Árbol (grafo) 8
II.4 Base de datos 9
II.5 Base de datos bibliográficas 9
II.6 Metadatos 9
II.7 Principio GIGO 9
II.8 Python 10
II.9 Flask 10
II.10 NetworkX 10
II.11 MongoDB 11
II.12 JSON 11
II.13 Pandas 11
II.14 Literatura científica 12
II.15. Publicaciones 12
II.16 Abstract (resumen) 13
II.17 Palabras claves 13
II.18 Taxonomía 13
CAPITULO III. Metodología 14

Descripción general de la metodología 14
Justificación de la metodología 14
CAPÍTULO IV. Desarrollo 16
Introducción 16
Primera iteración 17
Objetivos 17
Riesgos y alternativas 17
Desarrollo 17
Planificación 20
Segunda iteración 20
Objetivos iteración 20
Objetivos específicos relevantes 20
Riegos y alternativas 20
Desarrollo 21
Planificación 24
Tercera iteración 24
Objetivos específicos relevantes 24
Riegos y alternativas 25
Desarrollo 25
Planificación 28
Cuarta iteración 28
Objetivos iteración y objetivos específicos relevantes 28
Riegos y alternativas 28
Desarrollo 29
Búsqueda de módulos 29
Planificación 32
Quinta iteración 33
Objetivos 33
Riegos y alternativas 33
Desarrollo 33
Planificación 35
Sexta iteración 35
Objetivos 35
Riesgos y alternativas 36
Desarrollo 36
Planificación 43
Séptima iteración 44

Objetivos 44
Riesgos y alternativas 44
Desarrollo 44
Planificación 46
Octava iteración 46
Objetivos 46
Riesgos y alternativas 47
Desarrollo 47
Novena iteración 48
Objetivos iteración y específicos relacionados 48
Riesgos y Alternativas 48
Desarrollo 49
Planificación 55
Décima iteración 55
Objetivos iteración 55
Riesgos y alternativas 55
Desarrollo 55
CAPÍTULO V. Resultados 58
CAPÍTULO VI. Conclusiones y Recomendaciones 60
Conclusiones 60
Recomendaciones 60
REFERENCIAS BIBLIOGRÁFICAS 62
ANEXOS 64
Extracto de clasificación de palabras claves 64
Extracto de taxonomía IEEE CS 66
Thesis questionary 67
General instructions before answering: 67
Question set 1: 67
Question set 2: 67
Thesis questionary 68
General instructions before answering: 68
Question set 1: 68
Question set 2: 68

ÍNDICE DE TABLAS

Tabla 1: Similitudes entre MongoDB y MySQL 11
Tabla 2: Relación entre iteraciones y objetivos específicos 16
Tabla 3: Iteración 1 – Tecnología a usar 19
Tabla 4: Iteración 1 - Atributos utilizados para cálculo de métricas en bibliotecas digitales .. 20
Tabla 5: Iteración 2 - Tabla comparativa de atributos de artículos científicos por bases de datos 22
Tabla 6: Iteración 2 - Tabla comparativa de atributos de publicaciones científicas por bases de datos 22
Tabla 7: Iteración 2 - Tabla comparativa de formato de datos de bases de datos 22
Tabla 8: Iteración 3 - Comparación de funcionalidad de drivers entre R y python 26
Tabla 9: Iteración 4 - Tecnología a usar 28
Tabla 10: Iteración 5 - Tecnología a usar 29
Tabla 11: Iteración 5 - Formatos de exportación y lectura entre librerías NetworkX e igraph 30
Tabla 12: Iteración 4- Descripción de relación entre tipos de nodos 30
Tabla 13: Iteración 4 - Atributos de los nodos 32
Tabla 14: Iteración 4 - Atributos de las aristas entre nodos 31
Tabla 15: Repeticiones de palabras claves por tema Bioinformática 40
Tabla 16: Palabras claves organizadas por particularidad sin suavizar 42
Tabla 17 - Iteración 7: Coeficientes de tipos de publicación científica 45
Tabla 18: Iteración 9 - Tecnología definitiva a usar 50
Tabla 19: Palabras claves buscadas por los evaluadores según área de investigación 56
Tabla 20: Referencias a páginas e iteraciones donde se desarrollaron objetivos del TEG ... 58

ÍNDICE DE FIGURAS

Figura 1: Sección sub-grafo Bioinformática 37
Figura 2: Palabras comunes. Sección superior derecha del sub-grafo Bioinformática 38
Figura 3: Palabras Intermedias. Sección intermedia del sub-grafo Bioinformática 38
Figura 4: Palabras menos comunes. Sección inferior del sub-grafo Bioinformática 39
Figura 5: Fórmula para calcular particularidad de una palabra clave, versión sin suavizar ... 41
Figura 6: Fórmula para calcular particularidad de una palabra clave, versión suavizada 42
Figura 7: Particularidad suavizada vs sin suavizar 43
Figura 8: Fórmula para calcular particularidad de una obra científica 45
Figura 9: Visualizaciones candidatas 51
Figura 10: Interfaz – Inicio 53
Figura 11: Interfaz – Resultados 53
Figura 12: Interfaz - Detalle de resultados 54

SIPNOSIS

En el área de la Bibliometría, se han creado métricas tales como índice h, índice i, impact factor, normalmente basadas en los metadatos número de citaciones, bibliografía, fecha de publicación, orden de autores, palabras claves, que se utilizan en la búsqueda en las bases de datos bibliográficas para mostrar en formato de lista los artículos de mayor impacto científico y más relevantes dados unos términos de búsqueda. No obstante, estas métricas y los filtros principalmente ofrecidos en la búsqueda, autores, fecha de publicación o relevancia (según términos de búsqueda) no son particularmente útiles desde un punto de vista científico. Por lo que la investigación sigue siendo una tarea que requiere de grandes cantidades de tiempo y esfuerzo. Adicionalmente, los resultados en formato de lista no permiten saber la relación de los resultados entre sí, este formato solo representa un orden de relevancia respecto a los términos de búsqueda. Más aún, la disponibilidad y cantidad de literatura científica aumenta cada vez más, por lo que la idea de leer cada artículo disponible para investigar es poco factible.

Por lo tanto, se propone un algoritmo de minería de datos bibliográficos basados en que muestre un conjunto de resultados ordenados en jerarquía para ayudar en la investigación.

La evaluación de los resultados del algoritmo y la herramienta de visualización, mediante cuestionarios a dos profesores en dos áreas de investigación de su especialidad, fueron calificados como generalmente positivos y que valen la pena seguir desarrollando.

INTRODUCCIÓN

Las capacidades de búsqueda de bases de datos bibliográficas disponibles (Biblioteca Digital de ACM[1], IEEE Xplore[2], PubMed[3], etc) actualmente pueden ofrecer, principalmente, un conjunto plano de resultados que se puede ordenar por diferentes criterios (por orden alfabético, por autores, o por fecha de publicación o cuánta relevancia tiene un documento según los términos de búsqueda) que no son particularmente útiles para un investigador desde un punto de vista científico.

El presente trabajo especial de grado (TEG) trata sobre un algoritmo de minería de datos para identificar las publicaciones científicas según relevancia y ámbito en áreas del conocimiento. Dadas las limitaciones no se realizó un trabajo en todas las áreas del conocimiento, pero se ofrece una solución lo suficientemente general como para poder ser aplicada luego de aplicar procedimientos

Los principales aportes de este TEG son: Métrica para medir alcance de palabras claves y literatura científica, respecto a un área de investigación; algoritmo para ordenar jerárquicamente literatura científica según la métrica; y una interfaz que presenta de forma jerárquica las obras científicas.

[1] http://dl.acm.org/
[2] http://ieeexplore.ieee.org/Xplore/home.jsp
[3] http://www.ncbi.nlm.nih.gov/pubmed

CAPÍTULO I. Problema

I.1. Planteamiento del Problema

El incremento de la cantidad de artículos científicos disponibles por años ha crecido significativamente y de forma sostenida. Esto hace que mantenerse al día leyendo todas las publicaciones respecto a un tema una idea poco factible dada la cantidad de tiempo que se debe invertir.

Adicionalmente, los motores de búsqueda especializados para la investigación científica actualmente no permiten ordenar los resultados de una consulta respecto a un tema según un orden determinado para leer, lo cual hace la investigación más difícil y longeva. Aunque permite ordenarlos por calidad y por relevancia de las publicaciones con los términos de búsqueda.

Para permitir a estudiantes, docentes e investigadores investigar con mayor facilidad, se propone un algoritmo de minería de datos que presente, en un orden jerárquico, publicaciones científicas, tomando en cuenta la calidad y ámbito.

I.2. Objetivos

I.2.1. Objetivo General

Diseñar e implementar un algoritmo de minería de datos que proporcione un conjunto jerárquico de resultados que ayuden al usuario en la elección de un grupo adecuado de artículos científicos para leer en un tema de investigación en particular.

I.2.2. Objetivos Específicos

1) Diseñar e implementar un modelo basado en grafos capaz de representar diferentes tipos de relaciones entre un conjunto de artículos científicos devueltos por una base de datos bibliográfica.
2) Evaluar y seleccionar una base de datos de metadatos.
3) Diseñar e implementar mecanismo de acceso de datos adecuado a los metadatos.
4) Definir una métrica para evaluar el alcance y la relevancia de un artículo científico
5) Diseñar e implementar un algoritmo de minería de datos capaz de analizar el grafo bibliográfico para identificar los nodos más relevantes (artículos científicos) y ordenarlos jerárquicamente por relevancia para un tema de investigación determinado.
6) Diseñar e implementar una interfaz de usuario web simplificada para ejecutar el algoritmo de minería de datos
7) Integrar el mecanismo de acceso de datos, el algoritmo de minería de datos, y la interfaz de usuario.
8) Realizar la evaluación de la eficacia del algoritmo de minería de datos

I.3. Alcance

La métrica y el algoritmo se basarán en los metadatos disponibles para cada artículo almacenado en la base de datos destino y un enfoque basado en grafos se utilizará para representar las relaciones entre los artículos científicos, y para comprender mejor su rol (de los artículos científicos) en el contexto de un tema de investigación científica. El resultado del algoritmo de minería de datos de artículos científicos será una conjunto jerárquico (ordenado por relevancia y alcance) que un investigador puede utilizar para comprender mejor un tema de investigación en particular.

Los criterios utilizados para identificar la base de datos bibliográfica (con de metadatos de artículos científicos) a utilizar, teniendo en cuenta que el objetivo no es evaluar en general la calidad las bases de datos de metadatos de artículos científicos disponibles o modelos de redes bibliográficas, sino sólo evaluarlos para el uso en el contexto de esta Trabajo Especial de Grado, será:

- La calidad de los datos.
- La disponibilidad de los datos.
- Cuán relevante son los metadatos para sugerir más material de lectura sobre un tema similar.
- La interfaz de acceso a datos que las bases de datos disponibles.

La tecnología a utilizar en el Trabajo Especial de Grado (el lenguaje de la aplicación del algoritmo, mecanismo de acceso a datos, interfaz de usuario, etc) todavía no está definida y es parte del desarrollo del Trabajo Especial de Grado.

La eficacia del algoritmo se evaluará preguntando a personas, que ya conocen un tema de investigación, que llenen un cuestionario diseñado para evaluar los resultados del algoritmo. Esto implica la creación del cuestionario, ponerse en contacto con un grupo de investigadores familiarizados en los temas de investigación que estén en la base de datos destino, y pidiéndoles que evalúen

los resultados del algoritmo al completar el cuestionario. También implica que los resultados de estos cuestionarios se compilarán, analizarán y se presentarán conclusiones. Además, los cuestionarios respondidos se anexarán en el Trabajo Especial de Grado.

I.4. Limitaciones

A menudo, no todos los metadatos están disponibles para todos los artículos científicos en una base de datos bibliográfica. Usualmente los artículos científicos viejos tienen muchos menos metadatos vinculados a ellos. Esto, obviamente, afectará la fiabilidad del resultado final. Sin embargo, este problema desaparecerá con los artículos científicos más recientes, que en todo caso son de mayor importancia para los estudiantes e investigadores.
Las pruebas del algoritmo de minería de datos no involucrará todas las áreas de conocimiento, sino sólo una o dos áreas que sean posibles dada la base de datos destino. El algoritmo se realizará para que funcione con áreas del conocimiento en general. Las áreas no se pueden considerar a priori porque se está sujeto a la disponibilidad de los expertos, la calidad de los datos según el área, y, por último, pero no menos importante, para no reducir el alcance del Trabajo Especial de Grado.

Las técnicas de minería de datos a utilizar serán parte del alcance del Trabajo Especial de Grado. Elegir una a priori sin investigación supondría un riesgo muy alto y le reduciría alcance al Trabajo Especial de Grado.

No será extraído texto directamente de los artículos científicos, sólo sus metadatos. Si la base de datos seleccionada almacena y permite el acceso a su resumen (abstract) será considerado en el algoritmo.

El número de expertos a realizar la prueba del algoritmo, para evaluarlo, para cada área de conocimiento serán mínimo dos. Es decir, para cada área se elegirán al menos dos expertos.

I.5. Justificación

El resultado de este Trabajo Especial de Grado, puede ser una contribución importante para ayudar a hacer un mejor uso del conocimiento científico que ya está disponible en la web. La enorme cantidad de información que se acumula cada día en bases de datos públicas es inútil si no se apoya en técnicas de minería de datos que permitan la extracción de nueva información a partir de estos datos. En varios campos ya se utiliza este enfoque, por ejemplo, en la Biología, la Economía, la Meteorología y la Astronomía. Por algunas razones los datos bibliográficos son unos de los conjuntos de datos con mayor disponibilidad, pero entre las que menos se dirigen esfuerzos en las comunidades científicas de minería de datos.

Si resulta positivo el producto realizado, ésta podría ser una herramienta muy útil para los estudiantes e investigadores de todo el mundo.

CAPÍTULO II. Marco Teórico

II.1 Minería de Datos

La minería de datos "es el proceso del descubrimiento de información" (...) a este proceso también se le llama "« descubrimiento de conocimiento », el cual describe mejor este proceso" (Han, Pei, & Kamber, 2011). La finalidad del «descubrimiento de conocimiento» es el de extraer información útil y comprensible a partir de bases de datos, data warehouses, u otros tipos de repositorios de información, para la toma de decisiones.

Este proceso generalmente se realiza en 7 pasos: Limpieza de datos, integración de datos, selección de datos, transformación de datos, minería de datos, evaluación de patrones, y representación de datos. (Han, Pei, & Kamber, 2011)

II.2 Grafos

Los grafos son una "construcción teórica (constructo) compuesta de vértices (también llamados nodos) conectados mediante aristas. Los grafos representan datos estructurados. Los vértices de un grafo simbolizan pedazos discretos de información, mientras que las aristas de un grafo simbolizan las relaciones entre esos pedazos". (Samatova, William, Jenkins, Padmanabhan, & Chakraborty, 2014) . Tanto los vértices como las aristas usualmente tienen etiquetas (información asociada) que permite descubrir patrones, o relacionarlas.

II.3 Árbol (grafo)

Un gráfo de tipo árbol está definido de la siguiente forma: "Un grafo es un árbol si y solo si es un grafo conectado sin ciclos" (Samatova, William, Jenkins, Padmanabhan, & Chakraborty, 2014). Es decir, un grafo que no tenga alguna

arista de un nodo que haga trayectoria cerrada (circuito) con algún otro nodo es un grafo.

II.4 Base de datos

Las bases de datos son "un conjunto de datos relacionados entre sí que tienen un significado para alguien o algo". (Cardoso, 2006). Los datos que la conforman "generalmente están estructurados de forma en que puedan ser buscados y accedidos automáticamente" (Feather & Sturges, 2003).

Esto hace que las bases de datos sean "un conjunto de información sistemáticamente ordenados" (Feather & Sturges, 2003)

II.5 Base de datos bibliográficas

Las bases de datos bibliográficas son un tipo de bases de datos que "contiene referencias a literatura publicada, incluyendo papers, reportes, libros, actas de congresos, etc." (Feather & Sturges, 2003). En contraste a catálogos en línea que contienen la literatura publicada en sí misma, en formato digital, éstas bases de datos describen la literatura (metadatos) mediante abstracts, palabras claves, u otra información útil para describir la información.

II.6 Metadatos

"Los metadatos son datos acerca de datos" (University, 2015). Esta tipo de información es "descriptiva sobre un conjunto determinado de datos, un objeto o recurso, incluyendo la forma en que está formateado, y cuándo y por quién fueron recogidos. Aunque los metadatos más comúnmente se refieren a los recursos web, puede ser sobre cualquier recurso físico o electrónico. Puede ser creado automáticamente usando el software o manualmente." (Association, 2015).

II.7 Principio GIGO

El principio GIGO (Garbage In - Garbage out) se define de la siguiente forma "no importa qué tan inteligente sea un algoritmo de minería de datos, fallará

en descubrir conocimiento si se aplica en datos de baja calidad" (Freitas, 2002). Se considerará este principio mediante el desarrollo del TEG, y se hará referencias a él cuando sea usado.

II.8 Python

Python es un lenguaje de programación interpretado, interactivo, orientado a objetos. Incorpora módulos, excepciones, tipos de datos dinámicos, tipos de datos abstractos y clases. (Foundation, 2015)

II.9 Flask

Flask es framework web modular, que provee los servicios web básicos y otras funcionalidades mediante extensiones. Las dependencias, y a su vez componentes, principales son Werkzeug y Jinja2. Werkzeug provee toda la funcionalidad de enrutamiento, debugging y subsistemas Web Server Gateway Interface (WSGI). Asimismo, Jinja2 provee la funcionalidad de programación y soporte de plantillas HTML. Cualquier otra funcionalidad debe ser habilitada mediante extensiones. (Ronacher, 2015)

La filosofía de modularidad de éste framework permite que se desarrolle con los componentes esenciales, y añadir funcionalidades según sea la necesidad. Esto apoya un desarrollo rápido de aplicaciones web, ya que no te tiene que aprender el funcionamiento interno del framework y la forma en que fue pensado el framework para encarar problemas.

II.10 NetworkX

"NetworkX es un módulo de Python para la creación, manipulación, y estudio de la estructura, patrones y funcionamiento de grafos complejos. Adicionalmente, se pueden cargar y almacenar grafos in formatos estándares y no estándares, analizar estructuras de redes, construir modelos de redes, diseñar nuevos algoritmos de grafos, dibujar redes, entre otras." (NetworkX - Overview, 2015)

II.11 MongoDB

MongoDB es una base de datos orientada a documentos. Los datos se almacenan individualmente en "documentos" y son agrupados por "Colecciones". Además, los campos entre cada documento pueden variar, lo cual hace que tenga un esquema dinámico. (The MongoDB 2.6 Manual, 2015). Algunos conceptos de MySQL tienen su analogía en MongoDB. La Tabla 1 muestra conceptos comunes y similares en ambas tecnologías.

Tabla 1: Similitudes entre MongoDB y MySQL

MySQL	MongoDB
Tabla	Colección
Fila	Documento
Columna	Campo
Joins	Documentos anidados

Fuente: https://www.mongodb.com/compare/mongodb-mysql

II.12 JSON

JSON (JavaScript Object Notation) es un formato ligero para el intercambio de datos. Está basado en dos estructuras: Una colección de parejas nombre/valor, similares a las estructuras de datos objeto, diccionario o struct. Y una lista ordenada de dichos nombres/valores, modelada como la estructura de datos lista. Dado que estas son estructuras de datos universales, prácticamente todos los lenguajes de programación pueden intercambiar información en JSON. (Crockford, 2015)

II.13 Pandas

"Es un módulo de Python que provee estructuras de datos rápidas, flexibles y de fácil uso, diseñadas para trabajar con datos relacionales o etiquetados de forma tanto fácil como intuitiva (...) para el análisis de datos práctico y de 'vida real'". (Pandas, 2015)

Pandas está basado en NumPy, una librería de cómputo científico madura. Sus estructuras de datos fundamentales son las Series y los DataFrame,

similares a las estructuras de datos vectores y matrices, las cuales tienen interfaces a librerías de cómputo científico para permitir y facilitar el análisis de datos. Estas características hacen de pandas un módulo de análisis de datos confiable, eficiente y rápido.

II.14 Literatura científica

De forma general, la literatura científica "es el medio principal por el cual los investigadores comunican los resultados de sus investigaciones". Estas obras constituyen una memoria permanente del conocimiento científico y a su vez un registro de progreso en la investigación científica. "Existen diferentes tipos de literatura científica, normalmente referidos como **primaria, secundaria, terciaria** y gris" (Schembri, 2007)

Las obras **primarias** "son las investigaciones realizadas por un investigador, o un grupo, las cuales son publicadas en una publicación científica que tengan un proceso de revisión por pares (peer-review), y exija que la obra cumpla con un formato" (Schembri, 2007). Los formatos varían según la publicación, pero normalmente exigen: título, resumen (abstract), palabras claves (keywords), resultados, referencias, etc. La literatura **secundaria** son los trabajos que se basan en obras primarias para realizar resúmenes o síntesis sobre un área determinada de investigación. Van dirigidas a investigadores que comprenden el área. (Schembri, 2007) . La literatura **terciaria** se "basa en obras primarias y secundarias, y están dirigidas a un público general o a investigadores de otras áreas". (Schembri, 2007)

II.15. Publicaciones

Las publicaciones son el principal canal de divulgación de literatura científica. Las publicaciones incluyen las revistas académicas y las conferencias, y sus diferentes tipos. Estos tipos incluyen, reportes publicados de las conferencias

(transacciones o procedimientos), conferencias regionales o internacionales, revistas académicas especializadas, revistas de ciencia, entre otras.

II.16 Abstract (resumen)

Un abstract es un "breve resumen de un artículo científico, tesis, revisión, actas de conferencias o cualquier análisis profundo de un tema en particular. Los abstracts normalmente son usados para ayudar al lector para entender el propósito del paper." (Blake & Bly, 1993)

II.17 Palabras claves

Las palabras clave son términos o frases cortas que permiten clasificar y direccionar las entradas en los sistemas de indexación y de recuperación de la información en las bases de datos de un manuscrito o área temática en particular. (González Tous & Mattar V, 2012)

II.18 Taxonomía

Una taxonomía "es un Sistema de organización de conocimiento, un conjunto de palabras que han sido organizadas para controlar el uso de términos utilizados en un área de conocimiento en un "vocabulario" para facilitar el almacenamiento y extracción de objetos de un repositorio o base de datos. Estos sistemas de organización usualmente son específicos a un dominio de conocimiento." (Hlava, 2012)

CAPITULO III. Metodología

Descripción general de la metodología

Para el desarrollo de este Trabajo Especial de Grado, se eligió el modelo de desarrollo en espiral. El desarrollo en espiral es: "un generador de modelo de procesos conducido por el riesgo" (Boehm & Hansen, 2000). Se utiliza para guiar equipos de ingeniería de software concurrentes, enfocados al desarrollo de sistemas, con múltiples stakeholders (partes interesadas). Tiene dos características distintivas principales. Una es el enfoque cíclico para incrementalmente obtener un mayor grado de definición y ejecución de un sistema mientras que se disminuye su grado de riesgo. La otra es un conjunto de hitos clave para asegurar el compromiso de los stakeholders y así lograr soluciones de sistemas viables y mutuamente satisfactorios. (Boehm & Hansen, 2000)

Los hitos claves que siempre se deben hacer incluyen, invariantemente: **(1)** determinar los objetivos, alternativas y limitaciones de la iteración; **(2)** evaluar las alternativas, identificar y resolver los riesgos; **(3)** desarrollar y verificar los entregables de la iteración; y **(4)** planificar la próxima iteración (Thayer & Boehm, 1988).

Justificación de la metodología

La elección de la metodología de desarrollo en espiral se debe principalmente a la naturaleza experimental de este Trabajo Especial de Grado. Aunque hay requisitos claramente definidos que deben cumplirse, y lo más probable es que no cambien, la tecnología, la interfaz de usuario, los campos de metadatos y el acceso a los datos no está definidos, son interdependientes y necesarios para el producto final. Por lo tanto, la

determinación de las limitaciones, alternativas, riesgos para cada decisión, y la planificación son clave para elegir o probar la mejor alternativa posible a fin de alcanzar el producto final. Por otra parte, el rediseño y adaptación constante para cada componente de software es muy probable que suceda a fin de que se integren, que mediante esta metodología es posible hacerlo rápidamente, debido a su enfoque cíclico. Por último, otro factor importante para elegir este modelo es la participación de las partes interesadas de este Trabajo Especial de Grado, que son el tesista y el tutor.

CAPITULO IV. Desarrollo

Introducción

El desarrollo fue realizado en un total de diez (10) iteraciones de la metodología espiral. Cada iteración tiene sus objetivos a cumplir y también se especifican cuáles objetivos específicos del TEG están relacionados. En cada iteración se especifican los 4 procesos básicos de cada iteración. Al finalizar las iteraciones, se cumple el objetivo general del TEG. La Tabla 2 representa la relación entre la iteración y los objetivos específicos relevantes.

Tabla 2: Relación entre iteraciones y objetivos específicos

Número de la iteración	Objetivos específicos relevantes
Primera	Estudio dominio del TEG
Segunda	2
Tercera	3
Cuarta	1 y 3
Quinta	1 y 3
Sexta	4, 5 y 7
Séptima	4
Octava	5 y 7
Novena	6 y 7
Décima	8

Fuente: Elaboración propia

Primera iteración

Objetivos

- Realizar una investigación de dominio del TEG para hacer una lista de ideas para considerarlas en la definición de la nueva métrica y otros aspectos de la tesis.
- Realizar una tabla sobre posibles tecnologías a usar en el TEG.
- Realizar una lista de los atributos que podría tener la Base de datos a elegir.

Riesgos y alternativas

Un posible riesgo es no tener acceso a la información del dominio del TEG. Tales como artículos científicos, libros, presentaciones, entre otros. Dado el caso, el tutor académico puede proveer el material u orientar en la investigación hacia medios disponibles.

Desarrollo

Estudio del dominio del problema e ideas extraídas

En (Schembri, 2007) se expone que la literatura científica se puede dividir en tres tipos: primaria, secundaria y terciaria. Esta separación puede ser útil en el desarrollo del algoritmo, ya que puede ayudar ya sea a ordenar o a formar una jerarquía en un conjunto de literatura científica. Por lo tanto, se considerará esto una posible idea a utilizar

También se considerarán las palabras claves, título y el resumen (abstract) de cada literatura científica, ya que estos metadatos clasifican, contextualizan y resumen el propósito de cada literatura científica. Dichos metadatos se consideran confiables, ya que depende los autores elaborarla. Los autores son responsables de su asignación y ellos conocen, con gran detalle, su trabajo elaborado y el dominio donde se inscribe.

Otra idea a considerar es usar la clasificación de la literatura en sí o de las publicaciones mediante taxonomías. Esto puede ayudar a identificar sus

alcances u orden en una jerarquía. En las taxonomías de ciencias de la computación tales como ACM, IEEE, INSPEC, está la idea de que las áreas de conocimiento están compuestas de conceptos anidados. Esta relación entre conceptos significa que aquellos estén más arriba en la jerarquía, son más generales, y aquellos que estén más abajo son más particulares. Los conceptos deben ser más generales para poder abarcar, con sentido, aquellos conceptos que están anidados. Un punto débil de este tipo de modelo de conocimiento es que un concepto puede estar clasificado entre más de un área, por lo que la redundancia al modelar varias áreas puede aumentar. Sin embargo, esto también compone una oportunidad para detectar confluencias para detectar posibles nuevas aplicaciones.

A esta instancia, esas serán las ideas principales en las que se basará la tesis para ordenar según ámbito y relevancia cada literatura científica.

Sugerencias tecnología

De la investigación inicial, se encontró que los lenguajes de programación más populares para la minería de datos en general son: R y Python (Piatetsky, 2014).

Python es un lenguaje de programación de propósito general, que es ampliamente utilizado para resolver una amplia cantidad de tareas y problemas. En la última década, su uso por científicos e ingenieros se ha incrementado debido a la madurez del ecosistema de librerías científicas (Pedregosa, et al., 2012). El lenguaje de programación R "es un conjunto integrado de software para la manipulación, calculo y visualización de datos", la cual contiene "una colección grande, coherente e integrada de herramientas para el análisis de datos". (Venables, Smith, & Team, 2004)

Para ambos lenguajes, existen muchas librerías de minería de datos y de análisis de grafos. Para el análisis de grafos, se encuentra la librería igraph disponible en ambos lenguajes. Para realizar cálculos estadísticos, se

encontró la librería Pandas, Scipy, Scikit-learn para Python, y para R sus librerías que vienen por defecto.

Respecto a la conexión a bases de datos, ambos lenguajes tienen disponibles drivers para bases de datos relacionales y no relacionales. Por lo tanto, se puede elegir bases de datos de alguno de estos tipos.

Asimismo, respecto a frameworks web, para llevar el producto realizado a la web, Python cuenta con una gran variedad de opciones. Se pueden utilizar: Web2py, Django, Flask, etc. Por otro lado, R tiene menos opciones, solamente Shiny y rApache. Así, ambos lenguajes de programación son considerados para el desarrollo del TEG. Un resumen de lo anterior se encuentra en la Tabla 3.

Tabla 3: Iteración 1 – Tecnología a usar

Lenguajes	Drivers de BD	Librerias de grafos	Frameworks web
Python	pymongo	igraph, networkX	django, web2py, flask
R	rmongodb, RMongo,mongolite	igraph	shiny, rApache

Fuente: Elaboración propia

Sugerencias campos de la base de datos

Se realizó una tabla resumen con los metadatos normalmente utilizados para el cálculo de métricas de bibliotecas digitales según (Yan, Huang, Tang, Zhang, & Li, 2012) y basados en los trabajos de la bibliografía (Pág. 62). Esta tabla servirá como guía para elegir la base de datos a utilizar en el TEG. La Tabla 4 especifica los atributos utilizados por literatura científica y por publicación científica.

Tabla 4: Iteración 1 - Atributos utilizados para cálculo de métricas en bibliotecas digitales

Atributos utilizados para cálculo de métricas en bibliotecas digitales									
Artículo científico	Título	Abstract	Palabras claves	Fecha de publicación	Citas a textos científicos	Citaciones del texto científico	Número de citaciones	Nombre y apellido de cada autor	Orden de autores
Publicación del artículo	Nombre	Reputación	Áreas de la publicación	Tipo de publicación	Categoría				

Fuente: Elaboración propia

Una vez cumplidos los objetivos de la iteración, se concluye y se planifica la segunda iteración.

Planificación

En la próxima iteración se buscará la base de datos a utilizar para el TEG. Se considerarán los criterios respecto a las limitaciones del TEG para elegirla.

Segunda iteración

Objetivos iteración

- Buscar, evaluar y seleccionar base de datos.

Objetivos específicos relevantes

2) Evaluar y seleccionar una base de datos de metadatos.

Riesgos y alternativas

El primer riesgo es no encontrar bases de datos con los metadatos que permitan implementar alguna de las ideas principales para medir alcance o algunos de la Tabla 4. Ante esta situación se podría considerar: complementar la información mediante API's externas o crear una base de datos de forma manual. De suceder, se consideraría alguna de esas posibilidades. El segundo riesgo es que los datos de todas las bases de datos no sean aptos para el uso de minería de datos. Aquí se aplica el principio GIGO ("garbage in – garbage out"). Por lo que se intentaría limpiar los datos primero; si esto no aumenta la

calidad de los datos, o no es posible, se considerará las opciones del primer riesgo. El tercer riesgo es no encontrar bases de datos en español. Ante esta eventualidad, se buscarán bases de datos en inglés. No obstante, no se considera una limitación grave no encontrar una base de datos en español, dado que el inglés es el idioma universal de la investigación; en muchas universidades hispanoparlantes requieren comprensión lectora del idioma inglés; y en muchas publicaciones científicas piden al menos título, palabras claves y abstract (resumen) en inglés para su indexación y localización. Finalmente, esta herramienta pudiera alcanzaría un mayor público si está hecha en inglés, y se seguiría resolviendo el problema planteado en el TEG.

Desarrollo

Buscar datasets

En primer lugar se buscaron bases de datos de metadatos bibliográficos de artículos científicos. Desafortunadamente, no se encontró ninguna base de datos bibliográfica pública y abierta en español, sino en inglés. Se listan a continuación las bases de datos encontradas:

- **DLBP**: **D**igital **B**ibliography & **L**ibrary **P**roject dataset[4].
- **Aminer** Scholar Data Challenge dataset[5].
- **IEEE C**omputer **S**ociety **D**igital **L**ibrary[6].

Luego de haber encontrado dichas bases de datos, se procedió a evaluarlos respecto a los criterios que fueron expuestos en el Alcance del TEG (pág. 4), y las ideas expuestas en la iteración anterior.

[4] http://dblp.uni-trier.de/xml/

[5] https://aminer.org/citation

[6] https://www.computer.org/csdl

Para facilitar la evaluación de las bases de datos, se hizo un total de tres cuadros comparativos. Uno de estos cuadros lista los atributos de artículos científicos que se consideraron importantes para su posible uso en el algoritmo, y otro cuadro con la misma finalidad, pero respecto a publicaciones científicas. Los atributos de los artículos científicos que se consideraron se encuentran en la Tabla 5 y para las publicaciones científicas en la Tabla 6. El Tercer cuadro (Tabla 7) lista las especificaciones técnicas por cada base de datos considerada. A continuación, las tablas referidas

Tabla 5: Iteración 2 - Tabla comparativa de atributos de artículos científicos por bases de datos

	Metadatos por artículo científico								
Bases de datos	**Título**	**Abstract**	**Palabras claves**	**Fecha de publicación**	**Citas a textos científicos**	**Citaciones del texto científico**	**Número de citaciones**	**Nombre y apellido de cada autor**	**Orden de autores**
IEEE CSDL	Sí	Sí	Sí	Sí	No	No	Sí	Sí	Sí
DBLP	Sí	Sí	No	Sí	No	No	No	Sí	Sí
Aminer	Sí	Sí	No	No	Sí	No	No	Sí	Sí

Fuente: Elaboración propia

Tabla 6: Iteración 2 - Tabla comparativa de atributos de publicaciones científicas por bases de datos

	Metadatos por publicación científica			
Bases de datos	**Nombre**	**Reputación**	**Áreas de investigación de la publicación**	**Tipo de publicación**
IEEE CSDL	Sí	No	Sí	Sí
DBLP	No	No	No	No
Aminer	No	No	No	No

Fuente: Elaboración propia

Tabla 7: Iteración 2 - Tabla comparativa de formato de datos de bases de datos

Nombre de Bases de datos	**Formato de Base de datos**	**Número de Publicaciones Científicas**	**Número de Literatura Científica**
IEEE CSDL	MongoDB	7.252	618.279
DBLP	XML	6.439	=~2.500.000
Aminer	Texto plano	0	2.244.021

Fuente: Elaboración propia

Respecto a la evaluación en sí, todas las bases de datos cumplen con el criterio de disponibilidad de datos, ya que se poseen sus dataset.

En segundo lugar, todas las bases de datos están en un formato que permite la implementación de una interfaz de acceso de datos. No obstante, con las bases de datos **DBLP** y **Aminer** se tendría que elegir e importar a un manejador de base de datos, mientras que con la **CSDL** ya se puede utilizar con el manejador de base de datos MongoDB.

Tomando la Tabla 5 para realizar las comparaciones entre las bases de datos, la **DBLP** es la que tiene la mayor cantidad de entradas, pero es la que tiene menor cantidad de metadatos para implementar las ideas planteadas en la primera iteración y para sugerir más material para leer. Entre la **CSDL** y **Aminer,** la **CSDL** es la base de datos con más metadatos que permite utilizar las ideas planteadas en la primera iteración. Además, entre estos metadatos, posee los que para este TEG se consideran de los más importantes, al tener sus artículos científicos los metadatos de palabras claves, título, abstract; y las publicaciones científicas tienen metadatos de tipo de publicación y áreas de investigación de la publicación.

Igualmente, basándonos en la tabla previamente mencionada, sólo la **Aminer** tiene citas de un artículo a otros. Esto permite calcular cuáles artículos científicos citan a un artículo, y a su vez calcular el número, el cual es una métrica importante para medir el impacto de un artículo científico, pero no el alcance, el cual es el objetivo del TEG.

Dado que la **CSDL** tiene los metadatos que permiten probar las ideas propuestas en la iteración anterior para calcular el alcance de una literatura científica, tiene una buena cantidad de entradas (7.252 publicaciones científicas con 618.279 artículos científicos), se tiene acceso a la totalidad de los datos y su interfaz de acceso es mediante el manejador MongoDB, se eligió como base de datos a utilizar.

Detalles estudio CSDL

Para obtener más información de la **CSDL** y evaluar la calidad (consistencia y uso acorde del campo) de sus datos, se utilizó la herramienta Variety[7]. A partir de esa herramienta se determinó que cada literatura científica está asociada con una publicación científica, es decir, cada literatura disponible es publicada y tiene solo una publicación científica asociada. Este atributo es issueId.

Cada publicación científica tiene un atributo que identifica el tipo de publicación que es. Este atributo es pubType y solo tiene cuatro (4) posibles valores para todas las 7.252 publicaciones. Los posibles valores de este atributo son: transacciones, revistas generales, cartas ('letters') y conferencias; lo cual permite hacer una equivalencia la determinación del tipo (primario, secundario o terciario) de literatura científica que está asociada a la publicación.

Respecto al atributo "áreas de investigación de la publicación" (researchArea), se observó que hay un total de aproximadamente 100.000 valores, de los cuales gran parte no corresponden con la finalidad del atributo. Por lo tanto, este campo no es de calidad y no se usará.

Planificación

Ya elegida la base de datos, se procederá a diseñar e implementar el mecanismo de acceso a datos. Para la próxima etapa se debe considerar que el driver del lenguaje sea compatible con MongoDB 2.6, puesto que es el formato de la base de datos elegida.

Tercera iteración

Objetivos específicos relevantes

3) Diseñar e implementar mecanismo de acceso de datos adecuado a los metadatos.

[7] Variety - https://github.com/variety/variety

Riesgos y alternativas

En esta iteración el principal riesgo es no poder realizar el mecanismo de acceso de datos debido a algún problema en los drivers para los dos lenguajes candidatos. La alternativa a este escenario es mudar la base de datos a un manejador de base de datos que sí tenga soporte a las funcionalidades requeridas o cambiar de lenguaje de programación.

Desarrollo

Elección de Drivers de la base de datos y lenguaje de programación

R tiene 3 posibles drivers para conectarse con una base de datos MongoDB: rmongoDB, mongolite y RMongo. Python solo tiene el driver pymongo. De estos cuatro drivers, sólo pymongo es recomendado[8] por MongoDB Inc., creadora de MongoDB. Python está divido en dos grandes versiones, versión 2 y 3 Se eligió Python versión 2 puesto que, a fecha de iteración, tiene más módulos para instalarle[9]. Es decir, se elige Python 2 para no limitar las opciones futuras de librerías.

Los aspectos a considerar en los drivers son:

- Índices y búsqueda de texto, porque sin esta funcionalidad la búsqueda por texto sucedería:
    - Las consultas serían altamente ineficientes
    - No se podrían utilizar lexemas (raíces de palabras en vez de palabras enteras) para mejorar la búsqueda de palabras claves.
    - No se podría utilizar métrica estándar de MongoDB para medir la relevancia de cada documento de la base de datos devuelto según consulta del usuario[10].

[8] https://docs.mongodb.com/ecosystem/drivers/python/
[9] https://wiki.python.org/moin/Python2orPython3
[10] https://docs.mongodb.com/v2.6/core/index-text/#index-text-text-search

- Compatibilidad con frameworks Aggregate y Map Reduce, ya que sin esta funcionalidad no se pueden hacer consultas agregadas, necesarias para agrupar y calcular a partir de los atributos.
- Devolución resultados de las consultas en formato de cursores, dado que mantener todos los documentos en memoria no es un acceso adecuado dado la gran cantidad de datos.
- Expresiones regex y lógicas, de lo contrario las búsquedas serían muy limitadas.

Se procedió a realizar pruebas de los los drivers según las funcionalidades que requiere el diseño. Estos resultados se recopilaron en un cuadro comparativo en la Tabla 8.

Tabla 8: Iteración 3 - Comparación de funcionalidad de drivers entre R y python

	Drivers para R			Drivers para python
Funcionalidades	**mongolite 0.8**	**Rmongo 0.0.25**	**rmongodb 1.8.0**	**pymongo 2.8**
Soporta funcionalidad de búsqueda simple	Sí	Sí	Sí	Sí
Soporta funcionalidad de búsqueda simple con regex y operadores lógicos	Sí	Sí	Sí	Sí
Soporta funcionalidad Regex y operadores lógicos en las búsquedas de texto	Sí	Sí	Sí	Sí
Soporta funcionalidad de MongoDB aggregate framework	Sí	Sí	Sí	Sí
Soporta parámetros opcionales para el aggregate framework	No	No	No	Sí
Soporta funcionalidad MongoDB Map reduce framework	Sí	No	No	Sí
Búsqueda con índices de texto	No	No	No	Sí
Opción de devolver cursores al hacer búsquedas	No	No	Sí	Sí
Opción de devolver cursores al hacer búsquedas mediante Map Reduce o Aggregate Framewok	No	No	No	Sí
Soporta resultados anidados al regresar búsquedas	Sí	Sí	Sí	Sí
Compatible con MongoDB v2.6	Sí	Sí	Sí	Sí
Soporta creación de índices de texto	No	No	No	Sí

Fuente: Elaboración propia

De la Tabla 8, se concluye elegir como lenguaje de programación Python con el driver Pymongo puesto que soporta todas las funcionalidades básicas.

Diseño e implementación de acceso de datos

Una vez elegido los drivers, se procedió a diseñar e implementar el mecanismo de acceso de datos. Se diseñó tomando en cuenta:

- Necesidad de usar parámetros de acceso a la base de datos. Dirección, puerto, nombre de base de datos, nombre de colecciones, etc.
- Posibilidad de añadir parámetros mediante los métodos de consulta a las colecciones. Por ejemplo, los parámetros de las consultas que permitan el uso del disco o de tablas auxiliares, si las agregaciones resultan muy grandes para ser contenidas en memoria mientras se ejecutan.
- Necesidad, de devolver resultados de las consultas como cursores, no como objetos en memoria. Porque MongoDB 2.6 no puede devolver resultados agregados mayores a 16MB, sino.
- Necesidad de crear índices de texto, con pesos dependiendo del campo, para optimizar la búsqueda de texto[11].
- Posibilidad de hacer interfaz de búsqueda de texto en forma automática, no la usada por defecto.

Una vez implementado lo diseñado, se obtuvo una versión inicial del mecanismo de acceso de datos. Éste mecanismo se encapsuló y se usa mediante la librería IEEE_Data.

Finalmente, la tecnología a usar definida por ésta iteración está indicada por la Tabla 9.

[11] https://docs.mongodb.com/v2.6/reference/operator/query/text/

Tabla 9: Iteración 4 - Tecnología a usar

Lenguaje	Drivers de BD	Librerias de grafos	Frameworks web
Python 2.7	Pymongo 2.8	Igraph 0.6.0, networkX 1.1	django, web2py, flask

Fuente: Elaboración propia

Planificación

Luego de haberse diseñado e implementado un mecanismo de acceso de datos inicial, ahora se procederá a elegir los módulos de modelado y minería de grafos para adaptar el driver, y poder formar el grafo para realizar la exploración de conocimiento.

Cuarta iteración

Objetivos iteración y objetivos específicos relevantes

1) Diseñar e implementar un modelo basado en grafos capaz de representar diferentes tipos de relaciones entre un conjunto de artículos científicos devueltos por una base de datos bibliográfica.

3) Diseñar e implementar mecanismo de acceso de datos adecuado a los metadatos.

Riesgos y alternativas

Existe el riesgo que usando el grafo propuesto en ésta iteración no presente alguna información útil, ya sea por el modelado o por mala interpretación, es decir, que el grafo no pueda brindar información o que no se pueda extraer información. En éste caso, se propondrá un grafo diferente.

Asimismo, como se tendrá que modificar el mecanismo de acceso de datos con los campos relevantes, y la población y algoritmos de la estructura de datos en la que se definió el grafo, está la posibilidad de que las librerías o el lenguaje de programación previamente elegidas (Tabla 9) no soporten estas modificaciones.

Desarrollo

Búsqueda de módulos

Dado que el lenguaje de programación es Python, y la versión es 2.7, se buscaron módulos compatibles y específicos para el tipo de minería de datos del TEG. Es decir, buscar módulos para el análisis de datos y texto, y el modelado y minería de grafos.

Módulos de grafos

Para el modelado y minería de grafos se consideraron las librerías NetworkX e igraph. Ambas librerías tienen una gran cantidad de funciones para la minería de datos, y de formatos de lectura y exportación[12] [13].Los grafos creados en estas librerías pueden ser manipulados si se usa un formato común. No obstante, sólo NetworkX exporta a un formato que es ampliamente usado para la web y librerías web, el formato JSON. Como los resultados del TEG deben mostrarse en una interfaz web, por lo tanto, se utilizará la librería NetworkX. Si es necesario una función de igraph, se convertirá a un formato común para finalmente retomar y mostrar los resultados en formato JSON. La Tabla 11 detalla los formatos de lectura y exportación de los grafos situación. La Tabla 10 muestra la tecnología a utilizar hasta esta iteración.

Tabla 10: Iteración 5 - Tecnología a usar

Lenguaje	Drivers de BD	Librerías de grafos	Frameworks web
Python 2.7	Pymongo 2.8	NetworkX 1.1	django, web2py, flask

Fuente: Elaboración propia

[12] http://igraph.org/python/doc/igraph.Graph-class.html
[13]https://networkx.github.io/documentation/networkx-1.10/reference/readwrite.html

Tabla 11: Iteración 5 - Formatos de exportación y lectura entre librerías NetworkX e igraph

	Formatos												
Librerías	GEXF	GML	Pickle	GraphML	JSON	LEDA	YAML	Sparse Graph 6	Pajek	DIMACS	DOT	NCOL	SVG
Network X 1.1	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No	No	No	No
Igraph 0.6.0	No	Sí	Sí	No	No	No	No	No	Sí	Sí	Sí	Sí	Sí

Fuente: Elaboración propia

Diseño y descripción general del grafo

En (Xiao-Li Li, 2009), se modeló el grafo de la siguiente forma: nodos como autores y las aristas entre nodos según la relación de coautoría. A los nodos se le asignan un peso según la calidad de la literatura científica hecha a lo largo de su carrera, y a las aristas según el orden de autoría en sus obras a lo largo de su carrera.

Inspirado en este modelo, en la clasificación de obras científicas de (Schembri, 2007), y diseñado para resolver el problema planteado por este TEG, el grafo en su totalidad está compuesto por palabras claves, literatura científica, y los lugares donde fueron publicados (publicaciones científicas) como nodos. Hay dos reglas por las cuales se rigen las aristas entre nodos del grafo. Entre publicaciones científicas y palabras claves se crea una relación si la literatura dentro de las publicaciones tiene esa palabra clave. Entre un nodo palabra clave y un nodo literatura científica se crea una arista si una literatura científica tiene esa palabra clave como miembro de su lista de atributos. La Tabla 12 ilustra estas relaciones.

Tabla 12: Iteración 4- Descripción de relación entre tipos de nodos

	Presencia de relaciones en los nodos según tipo		
Tipos de nodos	Literatura cientifica	Palabras claves	Publicación cientifica
Literatura científica	No	Sí	Mediante P.C
Palabras claves (P.C)	Sí	No	Sí
Publicaciones	Mediante P.C	Sí	No

Fuente: Elaboración propia

Al grafo también se le añadió la posibilidad de relacionar nodos mediante coincidencia difusa de palabras claves ('fuzzy matching'), en vez de coincidencias exactas, mediante métodos de distancia Jaro-Winkler y Damerau–Levenshtein para cumplir el objetivo de diseñar un grafo capaz de modelar diferentes tipos de relación que pueden existir entre nodos (objetivo específico 1) y su posible uso en el algoritmo. Ver Tabla 13.

Tabla 13: Iteración 4 - Atributos de las aristas entre nodos

	Atributos de relaciones en los nodos según tipo		
Tipos de nodos	**Existencia exacta de palabra clave**	**Similaridad "Damerau – Levenshtein"**	**Similaridad "Jaro–Winkler"**
Obras científicas	Sí	Sí	Sí
Palabras claves	Sí	Sí	Sí
Publicaciones	Sí	Sí	Sí

Fuente: Elaboración propia

Implementación del grafo

El grafo descrito fue implementado a nivel de datos de la siguiente forma: Dos listas de Python, una que contiene todos los nodos del grafo, la otra con todas las relaciones existentes entre los nodos. A su vez, estas listas tienen como elementos anidados diccionarios de datos, lo que permite asociar a cada elemento varios atributos. Se diseñó con estructuras básicas de Python para asegurar compatibilidad entre funciones nativas y librerías que las utilicen para no crear código redundante. Los atributos para los nodos son los indicados por la Tabla 14. Los atributos para las relaciones son los indicados por la Tabla 13:

Tabla 14: Iteración 4 - Atributos de los nodos

	Presencia de atributos en los nodos según tipo		
Atributos en los nodos	**Obras cientificas**	**Palabras claves**	**Publicaciones**
NodoID	Sí	Sí	Sí
MongoID	Sí	Sí	Sí
Tipo nodo	Sí	Sí	Sí
Tipo publicación	Sí	Sí	No
Título publicación	Sí	No	No
DOI	Sí	No	No
Número de citas	Sí	No	No
Abstract	Sí	No	No
Palabras claves	Sí	No	No
Número palabras claves	Sí	Sí	No

Fuente: Elaboración propia

Para la implementación exitosa del grafo, inicialmente se hicieron pruebas con un número limitado de literatura científica, para probar si los atributos se estaban agregado correctamente. Se fue incrementando el número de literatura hasta llegar a todas las obras de la base de datos. Este incremento permitió verificar casos bordes existentes, tales como no existencia de datos, mismo atributo pero con diferente tipo de datos (posible en MongoDB, pero no de base de datos relacionables), entre otros detalles de implementación.

Con esas acciones se refinó la librería IEEE_Data. Se modificaron los atributos a acceder según tipo de datos, y las operaciones a realizar para asignarlos. Esto concluye la integración a este punto de la base de datos, el mecanismo y el grafo. De ser necesario, se modificaría en un futuro IEEE_Data y IEEE_Graph.

Planificación

Para la próxima iteración se crearán los subconjuntos de datos que representen un área de investigación.

Quinta iteración

Objetivos

1) Diseñar e implementar un modelo basado en grafos capaz de representar diferentes tipos de relaciones entre un conjunto de artículos científicos devueltos por una base de datos bibliográfica.

- Refinar librería de grafos

3) Diseñar e implementar mecanismo de acceso de datos adecuado a los metadatos.

- Crear subconjuntos de datos que representen un área del conocimiento.
- Limpiar subconjuntos
- Refinar el mecanismo de acceso de datos

Riesgos y alternativas

El principal riesgo es que no pueda crear un método general para la creación de subconjuntos. Ante esta eventualidad, se buscaría una base de datos que ya tuviera creados los subconjuntos o se crearían manualmente, buscando artículos científicos de autores con áreas de investigación ya claramente definidas.

Desarrollo

Creación de subconjunto y su modelado en grafo

Para la creación de subconjuntos de datos que representen un área de conocimiento se usó la taxonomía de la IEEE Computer Society, la cual es realizada por expertos de diversas áreas de la computación en general para usarla como referencia y clasificación de obras científicas. Una muestra de la taxonomía referenciada está en ANEXOS (pág. 64).

Se examinó la taxonomía junto con el tutor y se decidió trabajar en las áreas "H.2.8.a Bioinformática" y en "B.X.3 Tolerancia a fallos en Hardware". Dado

que el tutor académico tiene amplia experiencia en dichas áreas[14][15], se decidió modelar éstas áreas. Esto ayuda a que su evaluación sea más precisa. A su vez, esto permite implementar el algoritmo en más de un área de conocimiento, para así cumplir las limitaciones del TEG.

Para crear los subconjuntos en sí mismos se sigue el siguiente procedimiento:

1) Se buscó en la Taxonomía IEEE CS la sección donde aparece "Bioinformática". Se creó una lista de palabras con la palabra "bioinformática" y las palabras más generales que a su vez abarcan ésta área. Es decir, se aplica la idea de Taxonomía descrita en pág. 18.
2) Se utilizó la lista de palabras anterior en una consulta sobre los atributos de cada literatura científica: título, abstract y palabras claves. La consulta se hizo mediante el índice de texto de la colección que contiene las obras científicas.
3) De la base de datos, se consideraron solo aquellas obras científicas que tuvieran palabras claves y títulos. Esto es para cumplir con el principio GIGO.

Una vez obtenido el subconjunto de datos del área Bioinformática, se procedió a modelar dicho subconjunto en grafo. De esta forma, se delimitó la base de datos a un área de investigación.

Como para el proceso de delimitación de subconjuntos existe un marco de referencia confiable (la Taxonomía IEEE CS) y aplicable a varias áreas de investigación, así como unos pasos reproducibles para diferentes áreas; se considera lo suficientemente general como para aplicarlo a otras áreas del conocimiento.

[14] https://scholar.google.com/citations?user=qf0BXToAAAAJ&hl=fr
[15] http://porto.polito.it/view/creators/Benso=3AAlfredo=3A002904=3A.html

Finalmente, se aplicó el procedimiento anteriormente descrito con para crear el área B.X.3 - Tolerancia a fallos en Hardware.

Detalles de implementación

Una vez obtenidos y almacenados ambos subconjuntos en colecciones independientes, se refinó la librería IEEE_Data para proveer acceso. Esto involucró crear sus respectivos índices de texto para cada colección, etc. No hubo cambios en IEEE_graph porque se siguen utilizando los mismos atributos. No se realizaron subconsultas ni se operó sobre una vista de la colección de literatura científica para representar el subconjunto, porque MongoDB v2.6 no soporta dicha funcionalidad, común en las bases de datos relacionales.

Planificación

Obtenidos los subconjuntos y los grafos que las representan, se comenzará a buscar patrones en los grafos.

Sexta iteración

Objetivos

4) Definir una métrica para evaluar el alcance y la relevancia de un artículo científico.

a. Diseñar e implementar métrica

5) Diseñar e implementar un algoritmo de minería de datos capaz de analizar el grafo bibliográfico para identificar los nodos más relevantes (artículos científicos) y ordenarlos jerárquicamente por relevancia para un tema de investigación determinado.

7) Integrar el mecanismo de acceso de datos, el algoritmo de minería de datos, y la interfaz de usuario.

b. Usar modelo de grafo anterior a las nuevas colecciones
c. Buscar patrones en las nuevas colecciones mediante su representación en grafos

Riesgos y alternativas

El principal riesgo es que el patrón en esta iteración no permita determinar el alcance de una obra, ya sea por la forma en que está modelado el grafo o alguna otra razón. De suceder esto, se buscaría otros patrones en el mismo subconjutno y grafo. Si tras intentar un número de veces no se encuentra un patrón satisfactorio, se consideraría elegir modelar un grafo distinto, buscar otra fuente de datos, o complementar la actual.

Desarrollo

Búsqueda de patrones

Con el subconjunto de Bioinformática y con la librería IEEE_graph, se creó el modelo de grafo descrito en la Cuarta iteración (pág. 28). En el grafo, se inició la búsqueda de patrones, específicamente en el subgrafo que comprende los nodos palabras claves y los nodos literatura científica para empezar a definir una métrica que ayude a identificar el alcance de una obra científica.

Inicialmente, se visualizó dicho subgrafo en diferentes disposiciones, tales como circular, concha y resorte (según implementación de Fruchterman-Reingold). De las disposiciones probadas, la disposición en resorte, respecto a los nodos palabras claves fue la más útil. Cuando ésta disposición se programó de tal forma que los nodos palabras claves con más aristas compartidas con nodos literatura científica eran los que más atraían a los nodos conectados se observó que aquellos nodos tipo palabras clave que tuvieran más aristas hacia nodos tipo literatura científica, parecían ser las palabras más generales del tema. Mientras que aquellas palabras claves que atraían pocos nodos literatura científica parecían ser más específicas. Por ejemplo en la **Figura 1: Sección sub-grafo Bioinformática**, donde los nodos azules son obras científicas y las amarillas son palabras claves, se observa que en la esquina superior derecha se encuentran las palabras claves comunes, con mayor cantidad de nodos literatura científica atraídos; en la

esquina inferior izquierda palabras claves menos comunes y con menos nodos literatura científica atraídos; y entre estas secciones, palabras claves 'intermedias'. Esta 'categorización' es particularmente útil, ya que a priori no se sabe cuáles son las palabras claves 'específicas' o 'generales' de un área, y ofrece una posibilidad de asignar a cada palabra un peso.

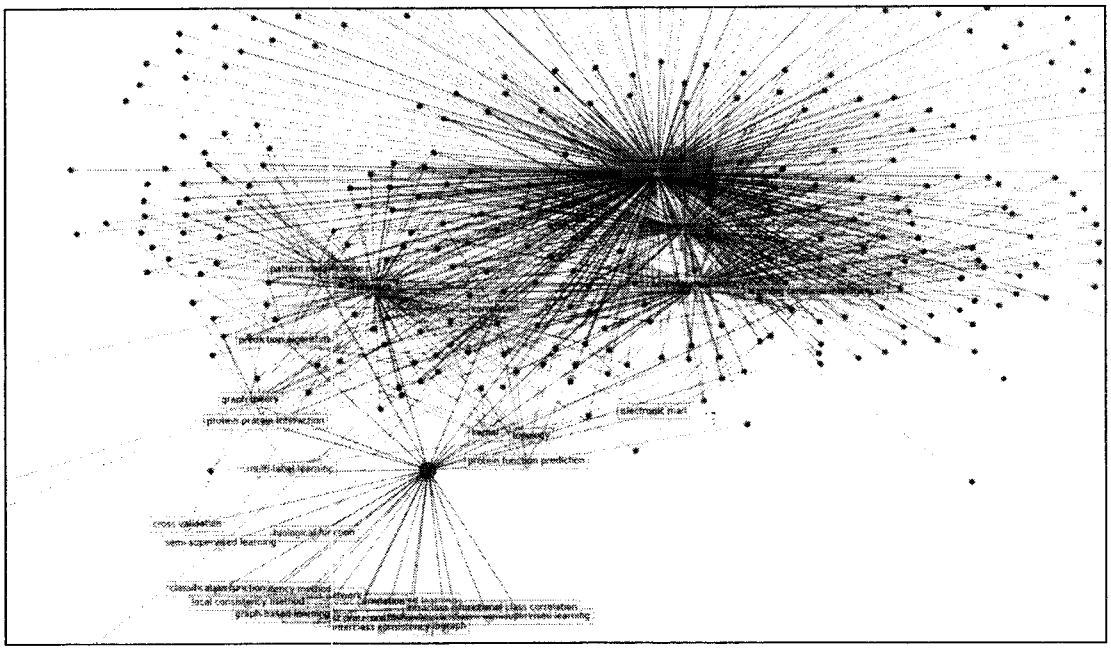

Figura 1: Sección sub-grafo Bioinformática

Leyenda: Nodos azules son obras científicas y las amarillas son palabras claves. Fuente: Elaboración propia

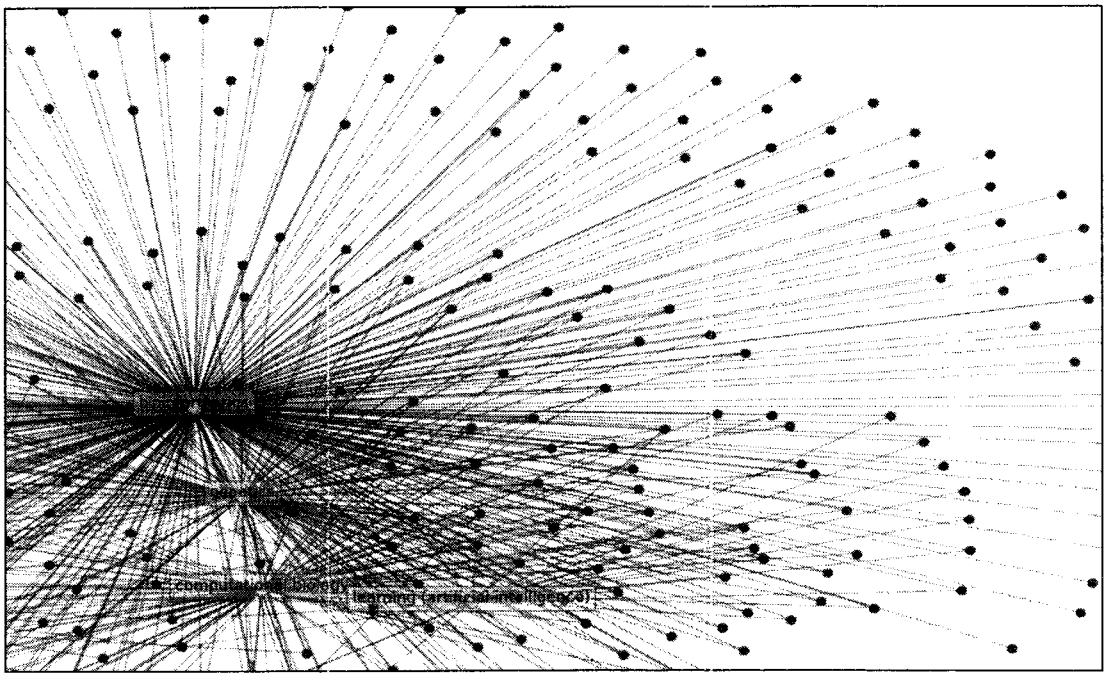

Figura 2: Palabras comunes. Sección superior derecha del sub-grafo Bioinformática

Leyenda: Nodos azules son obras científicas y las amarillas son palabras claves. Fuente: Elaboración propia

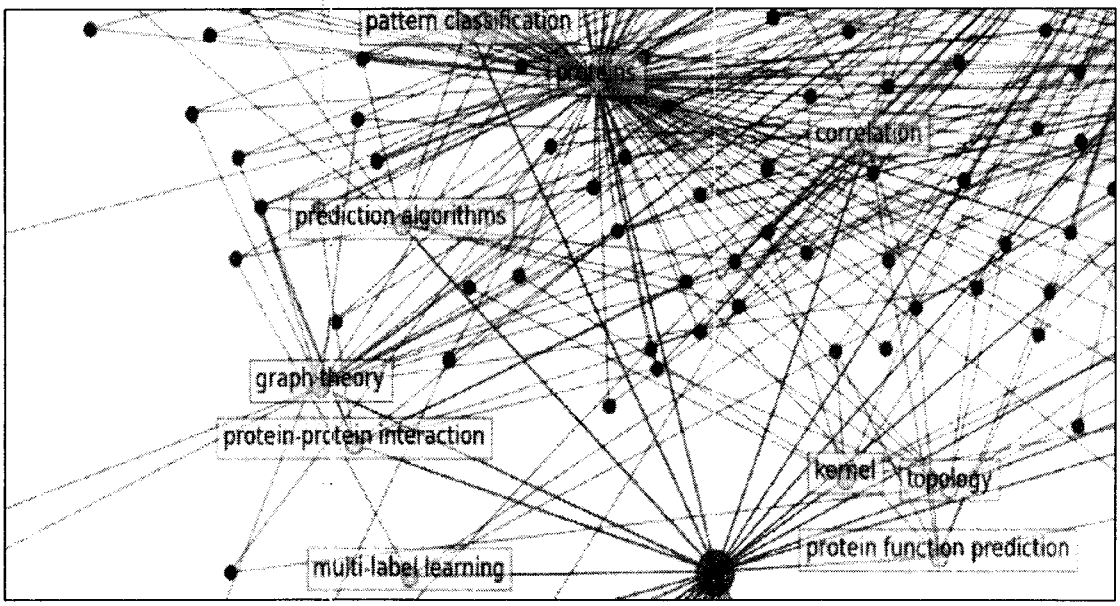

Figura 3: Palabras Intermedias. Sección intermedia del sub-grafo Bioinformática

Leyenda: Nodos azules son obras científicas y las amarillas son palabras claves. Fuente: Elaboración propia

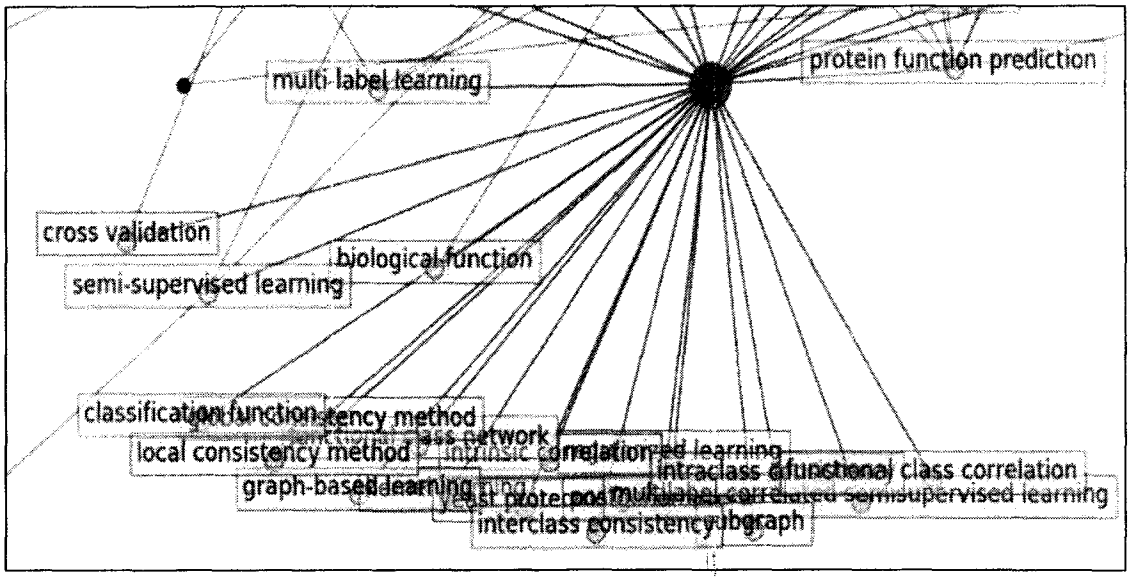

Figura 4: Palabras menos comunes. Sección inferior del sub-grafo Bioinformática

Leyenda: Nodos azules son obras científicas y las amarillas son palabras claves. Fuente: Elaboración propia

Posteriormente, se buscó profundizar sobre este **indicio** (forma válida de guiar un proceso de minería de datos según (Han, Pei, & Kamber, 2011) calculando la cantidad de aristas compartidas por cada palabra clave con cada obra científica. Se exportó a Excel los 500 nodos palabras claves con mayor cantidad de aristas compartidas con nodos literatura científica para su análisis. A ése cálculo se le referirá como ***repetición_tema(pc),*** y representa un porcentaje. Se ordenaron las palabras claves de mayor a menor porcentaje de repetición en la colección probada (Bioinformática). Algunos resultados en la Tabla 13.

palabra_clave	r_tema(pc)
bioinformatics	0.11624304
computational biology	0.03223807
ieee transactions	0.01136598
molecular biophysics	0.01053937
biology and genetics	0.00588965
molecular configurations	0.00423641
genetic algorithms	0.00309981
complexity theory	0.00247985
parallel processing	0.00232486
entropy	0.00232486
encoding	0.00222153
predictive models	0.00196322
arrays	0.00196322
evolution (biological)	0.00175656
linear programming	0.00175656
simulated annealing	0.00165323
data structures	0.00165323
bioinformatics (genome or protein) databases	0.00160157
probabilistic logic	0.00154991
graphics processing units	0.00134325
markov processes	0.00129159
parameter estimation	0.0011366
structural bioinformatics	0.00103327
upper bound	0.00103327
parallel algorithms	0.00098161
robinson-foulds distance	0.00087828
design methodology	0.00087828
incomplete lineage sorting	0.00082662
protein complex prediction	0.00082662
regression analysis	0.00082662
data handling	0.00077495
unsupervised learning	0.00077495
monte carlo methods	0.00072329
reverse-complement	0.00067163
inference algorithms	0.00067163
graphics processing unit	0.00067163
labeled expression data	0.00061996
heuristic programming	0.00061996
deep coalescence	0.00061996
alignment	0.00061996
exact enumerative algorithm	0.0005683
closest substring problem	0.0005683
tree local similarity	0.00051664
coexpression	0.00051664
rna local similarity	0.00051664
differential coexpression	0.00051664

Tabla 13: Repeticiones de palabras claves por tema Bioinformática

Fuente 1: Elaboración propia

A partir de los resultados arrojados por este cálculo, se observó junto con el tutor que no es suficiente la repetición de palabras en un subconjunto para determinar qué tan general o particular es una palabra, porque las palabras de mayor repetición muestran las áreas generales de investigación, clasificaciones generales de algoritmos, pero no métodos y técnicas específicas del área con mayor valor.

Se siguió refinando el diseño e implementación de la métrica. Para tener la perspectiva de la importancia de cada palabra clave respecto a la base de datos, se calculó la cantidad de enlaces que hay entre cada palabra clave y cada publicación. A partir del cálculo anterior, se computó el número de publicaciones científicas diferentes en que está involucrada la palabra clave. A Esto último se referirá como ***repetición_publicaciones(pc)***.

Luego, se ordenaron descendientemente las palabras claves dadas por la fórmula ***Particularidad(pc)***, para determinar qué tan general o particular es una palabra clave.

$$Particularidad(pc) = \left(\frac{repeticion_tema(pc)}{repeticion_publicaciones(pc)} \right)$$

Figura 5: Fórmula para calcular particularidad de una palabra clave, versión sin suavizar

Fuente: Elaboración propia

Donde ***Particularidad(pc)*** es la razón entre la repetición de una palabra clave en el subconjunto tema (dada por ***repetición_tema(pc)***)y la repetición de una palabra clave en todas las publicaciones de la base de datos ***repetición_publicaciones(pc).*** La intención de esta fórmula es sugerir qué tan particular respecto a un tema es una palabra clave. Mientras mayor es el número resultante, más particular respecto a un tema se interpreta la palabra. Asimismo, mientras más pequeña más general se interpreta que es.

Se usa esta interpretación, ya que a medida que ***repetición_tema(pc)*** aumenta en el numerador, más frecuente, en el subconjunto es ***repetición_publicaciones(pc);*** y a medida que disminuye en el denominador, menos frecuente en las publicaciones en general es. De esta forma queda inicialmente definida la "generalidad"-"particularidad" de una palabra clave mediante ***Particularidad(pc)***, como una relación inversamente proporcional entre ***repetición_tema(pc)*** y ***repetición_publicaciones(pc)*** dado una palabra clave. Se notó que mejoró las palabras claves obtenidas al ordenarlas decrecientemente según la fórmula mencionada, ya que incluyen métodos y problemas por encima de palabras generales (Tabla 14).

palabra_clave	part_sin_sua(pc)
ieee transactions	1.79299409
evolution (biological)	1.10839016
robinson-foulds distance	1.10839016
molecular configurations	1.06927051
incomplete lineage sorting	1.04319074
computational biology	0.94614974
reverse-complement	0.84759248
labeled expression data	0.78239306
heuristic programming	0.78239306
deep coalescence	0.78239306
exact enumerative algorithm	0.71719364
closest substring problem	0.71719364
bioinformatics (genome or protein) databases	0.67372735
molecular biophysics	0.66503410

structural bioinformatics	0.65199421
tree local similarity	0.65199421
coexpression	0.65199421
rna local similarity	0.65199421
differential coexpression	0.65199421
sequence-structure matching	0.65199421
bioinformatics	0.62691751
biology and genetics	0.61939450
biological pathway knowledge	0.58679479
variation calling	0.58679479
chinese cabbage	0.58679479
gene translation	0.58679479
prior probability construction	0.58679479
extended burrows-wheeler transform	0.58679479
supertree	0.58679479
pedigrees	0.58679479
pan-genome indexing	0.58679479

conserved gene clusters	0.58679479
protein complex prediction	0.52159537
optimal bayesian classifier (obc)	0.52159537
protein-ligand binding sites	0.52159537
incomplete annotations	0.52159537
phenotype classification	0.52159537
np-hardness,satsolver, idp	0.52159537
clustering validity index	0.52159537
synthetic pathway generation	0.52159537
blood serum analysis	0.52159537
gene tree parsimony	0.52159537
string transformation	0.52159537
gene teams	0.52159537

Tabla 14: Palabras claves organizadas por particularidad sin suavizar

Fuente: Elaboración propia

Posteriormente, esta fórmula se refinó, ya que en la anteriormente usada, a medida que ***repetición_publicaciones(pc)*** se acerca a 0, sube bruscamente el resultado, por ser una hipérbola. Esto supone un problema ya que un ligero cambio causa una correspondencia muy alta. Para resolver este problema, se usaron los modelos de crecimiento y decrecimiento clásicos16. La fórmula Particularidad(pc) se refinó para que el crecimiento fuera suavizado. La fórmula utilizada está en la Figura 6. En la Figura 7 aparece una comparación entre ambas formas de representar la particularidad, donde el eje vertical (eje Y) es el número asignado a cada palabra clave, y en el eje horizontal (eje X) están las palabras que conforman el tema.

$$Particularidad(pc) = 2.5\log_{10}\left(\frac{repeticion_tema(pc)}{repeticion_publicaciones(pc)}\right) + 1.2$$

Figura 6: Fórmula para calcular particularidad de una palabra clave, versión suavizada

[16] https://people.richland.edu/james/lecture/m116/logs/models.html

Fuente: Elaboración propia

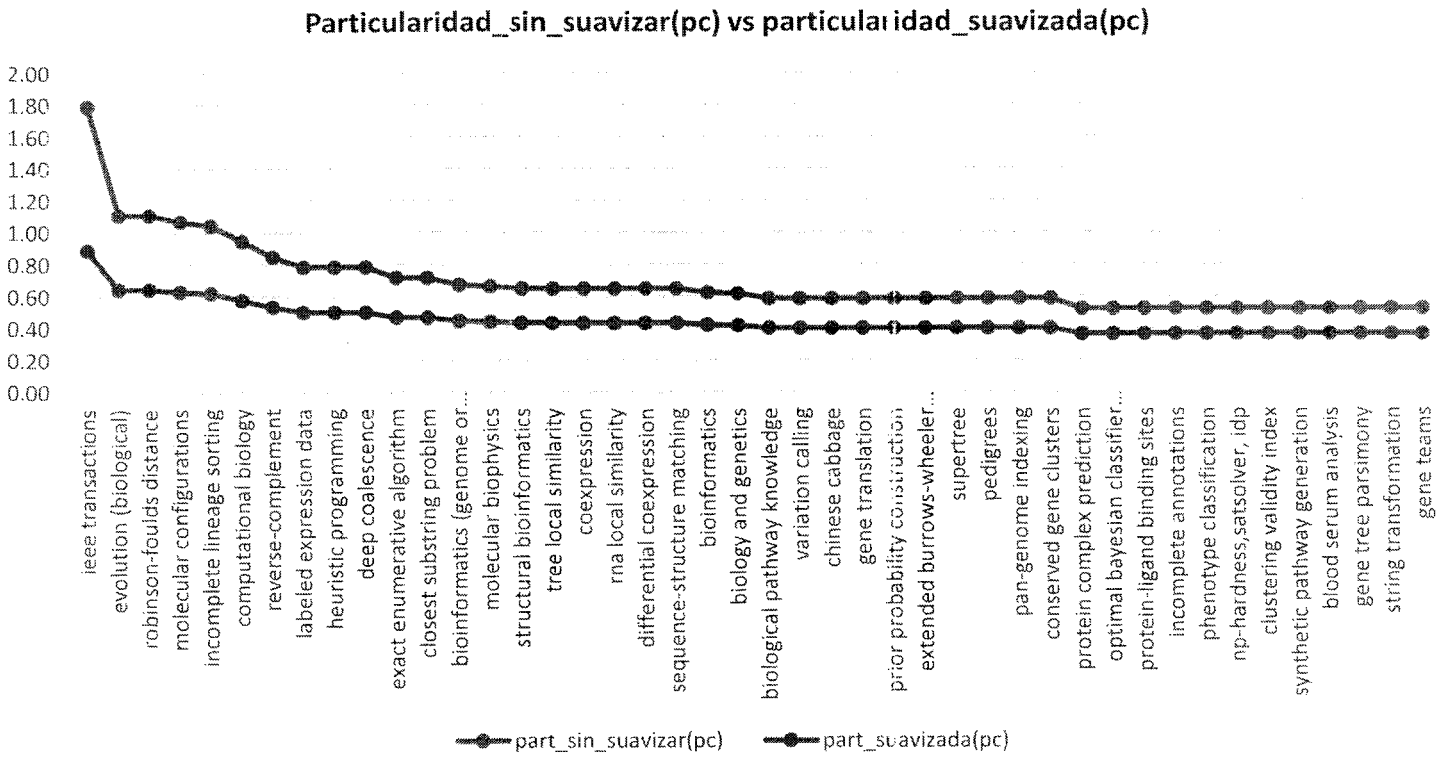

Figura 7: Particularidad suavizada vs sin suavizar

Fuente: Elaboración propia

Una vez definida la métrica por palabra clave ***Particularidad(pc)*** , se eligió la librería para realizar estos cálculos en Python en vez de usar Excel. La librería pandas fue seleccionada porque tenía las funciones y estructura de datos familiares con Excel, buen historial de usos profesionales previos, y total compatibilidad con la tecnología a usar. De esta forma, se hizo la programación de los cálculos en Python, para automatizar el proceso. Este proceso se encapsuló en la librería IEEE_Analysis.

Planificación

En esta instancia se diseñó e implementó parcialmente una métrica que mide por palabra clave su alcance según un área de investigación a partir de un patrón visual observado. No obstante todavía falta adaptar ésta métrica para

que mida el alcance y relevancia de una obra científica. A esta métrica inicial, se le incorporará atributos de literatura y publicaciones científicas.

Séptima iteración

Objetivos

4) Definir una métrica para evaluar el alcance y la relevancia de un artículo científico

a. Refinar métrica de la sexta iteración para medir el alcance y relevancia de una obra científica.

7) Integrar el mecanismo de acceso de datos, el algoritmo de minería de datos, y la interfaz de usuario.

b. Integrar consulta de usuario al sistema.

Riesgos y alternativas

No poder realizar desarrollar la métrica, que actualmente mide alcance y relevancia de una palabra clave, para medir alcance y relevancia a una obra científica, por falta de datos o por su mala calidad. Ante esta situación se buscaría complementar datos mediante API a partir del DOI de cada obra científica.

Desarrollo

Métrica para medir alcance de literatura científica

Se propone que para que la métrica mida el alcance de una literatura científica, se utilizarán sus atributos y los de su publicación asociada. Los atributos a usar son: el tipo de publicación correspondiente, y todas las palabras claves que tenga asociada.

A cada tipo de publicaciones disponibles (transacciones, revistas generales, cartas (letters), conferencias) se le asignó un coeficiente de particularidad. Se basó en la descripción hecha por tipo de publicación por la IEEE (IEEE, 2015) y la experiencia del tutor. Los resultados están en la Tabla 15. Se asignaron

de tal forma que mientras mayor sea el coeficiente del tipo de publicación, más específica es.

Tabla 15 - Iteración 7: Coeficientes de tipos de publicación científica

Tipo publicación	Coeficiente particularidad
Transacciones	1.5
Revistas generales	1.0
Cartas (letters)	1.6
Conferencias	1.4

Fuente: Elaboración propia y Alfredo Benso

Por cada obra científica se calculará la sumatoria de la ***Particularidad(pc)*** de cada palabra clave que le corresponda, divida entre su número de palabras claves. De esta forma, se compensa entre literaturas científicas que tengan menos palabras claves que otras, y se consideran todas las palabras que conformen una literatura científica. La fórmula viene dada según la ecuación en la Figura 8.

$$Particularidad_literatura(lit_cient) = \left(\frac{Particularidad(pc)}{N_pc(lit_cient)} \right) * tipo_publicacion(lit_cient)$$

Figura 8: Fórmula para calcular particularidad de una obra científica

Fuente: Elaboración propia

Integración de consulta de usuario con grafo y base de datos

Hasta ahora no se ha calculado la relevancia de los documentos respecto a los términos de búsqueda del usuario. Para esto, se utilizarán los índices de texto del manejador de base de datos MongoDB. Dicho índice funciona de la siguiente manera: MongoDB sólo asigna una puntuación a cada documento que contenga los términos de búsqueda en los campos indexados. Esta puntuación aumenta según la frecuencia de los términos de búsqueda, y según el peso asignado a cada campo. Es decir, en los documentos de la base de

datos, mientras más aparecen los términos en los campos con mayor peso, mayor será la puntuación, y por tanto, su relevancia.

Para el índice de texto de este TEG, se asignarán los siguientes pesos: 8 para el campo palabras claves, 4 para el campo títulos y 2 para el campo abstract. Esto equivale a que una ocurrencia de los términos de búsqueda en el campo palabras claves vale 3 veces. Por ejemplo, si en un documento la palabra 'protein' está en las palabras claves una (1) vez y en el abstract dos(2) veces, dicho documento tendrá una puntuación de (8*1)+(4*0)+(2*2) = 12.

Entonces, se procederá a utilizar solamente aquellos documentos que superen 2 de puntuación. De esta forma, se integrarán los términos de búsqueda del usuario, y utilizarán aquellos documentos que tengan una relevancia mayor a 2.

Finalmente, por alcance se entiende qué tanto cubre un trabajo científico; y **por relevancia** se entiende qué tan relevante es al término de búsqueda y al área de investigación.

Planificación

Una vez diseñada e implementada la métrica para medir la particularidad de cada literatura científica, y hecha la integración entre la consulta del usuario y la base de datos, se procederá a diseñar e implementar el algoritmo que analice los resultados obtenidos por la consulta del usuario en el área implementada. El algoritmo se basará en la métrica desarrollada en esta iteración.

Octava iteración

Objetivos

5) Diseñar e implementar un algoritmo de minería de datos capaz de analizar el grafo bibliográfico para identificar los nodos más relevantes (artículos científicos) y ordenarlos jerárquicamente por relevancia para un tema de investigación determinado.

7) Integrar el mecanismo de acceso de datos, el algoritmo de minería de datos, y la interfaz de usuario.

Riesgos y alternativas

Desarrollo

Una vez que el grafo bibliográfico y la métrica fueron hechas, se procedió a hacer el componente del algoritmo que ordena las obras científicas. El cual se puede dividir en dos partes, la elección de la primera obra científica a leer, es decir, la raíz del árbol, y la segunda, que la construcción de la jerarquía.

En la primera parte se elige cuál será la primera obra científica a leer, esta obra será referida como **Literatura Científica Inicial (LCI)**. Para esto, el algoritmo usa la métrica ***Particularidad_literatura(x)*** con aquellas literaturas científicas de que tengan al menos cuatro (4) palabras claves.

A partir de la obra **LCI**, se buscan en el grafo todas las otras obras que compartan sus palabras claves (aquí la importancia de restringir las obras con mínimo 4 palabras claves). Para la construcción de la jerarquía hacen los siguientes pasos:

Entre los nodos palabras claves en común entre la **LCI** y las obras identificadas, se crea una lista de grupos de combinaciones únicos de palabras claves. Se crear a partir de combinaciones únicas para no tener grupos duplicados. Ejemplo: [('Bioinformatica', 'genes'; 'Bioinformatica', 'dna'; 'Bioinformatica', 'dna', 'genes')], donde cada punto y coma separa un grupo, y cada coma un integrante del grupo.

Una vez obtenida esta lista de combinaciones, se agrupan los nodos según las combinaciones únicas presentes en la mencionada lista. En el caso de que pertenezca a más de una combinación única, se asigna la que tenga mayor cantidad de palabras claves en común.

Una vez asociadas las obras científicas en alguno de estos grupos, se le calcula a cada obra ***Particularidad_literatura(x)*** para determinar su alcance. Luego, se ordena descendientemente respecto a la métrica, por cada grupo. De esta forma, se encuentran los grupos claramente definidos, con obras asociadas y en orden. Para completar la jerarquía, se conectan las aristas entre sí y se conectan con el **LCI**.

Finalmente, se eliminan los nodos sobrantes del grafo, es decir, los que no quedan agrupados o no tienen palabras claves en común, y los nodos publicaciones y palabras claves. Así, se garantiza, que el grafo sea un árbol, es decir, se cumple la definición de un árbol. Durante el desarrollo de los pasos de éste algoritmo hasta esta iteración se graficaron los resultados mediante matplotlib, una dependencia de NetworkX, y se limitaron las obras científicas hasta llegar a la totalidad de la base de datos.

Novena iteración

Objetivos iteración y específicos relacionados

6) Diseñar e implementar una interfaz de usuario web simplificada para ejecutar el algoritmo de minería de datos.

c. Buscar, evaluar y elegir tecnología web para la interfaz web
d. Buscar, evaluar y elegir tipo de visualización jerárquica

7) Integrar el mecanismo de acceso de datos, el algoritmo de minería de datos, y la interfaz de usuario.

Riesgos y Alternativas

El principal riesgo es no encontrar librerías que sean compatibles con el formato de datos. Ante esta situación, se tendría que buscar y evaluar librerías que cumplen con los otros criterios en la sección Elección de tecnología web y adaptar el formato de datos del resultado para poder utilizarlas; en vez de elegir la librería basada en el formato.

Desarrollo

Planificación de la interfaz

La interfaz considerará principalmente tres aspectos. La visualización jerárquica de los resultados, la información complementaria de cada obra y la navegación de la jerarquía.

Elección de tecnología web para la visualización

Se buscaron librerías web que se pudieran utilizar para mostrar e interactuar con los resultados del algoritmo al usuario de forma jerárquica. Los criterios para elegir las librerías fueron: que sea compatible con formato JSON tipo árbol o grafo, que pueda mostrar resultados jerárquicamente, y que se puedan programar interacciones. Estos tres criterios son muy importantes. De la búsqueda realizada, se consideraron las librerías Cytoscape.js, D3.js y GraphViz. Los criterios para elegir la librería fueron los siguientes:

2) Compatible con formato JSON tipo grafo o árbol.
    a. D3.js: Sí es completamente compatible con JSON tipo árbol y grafo.[17]
    b. GraphViz: No es compatible por defecto con JSON[18]. Se necesitan librerías intermediarias para pasar de DOT (el lenguaje de GraphViz) a JSON.
    c. Cytoscape.js: Es compatible con formato JSON, pero con una estructura diferente a la actual[19].
3) Posibilidad de programar o usar interacciones web.
    a. D3.js: Las interacciones se deben programar. El sitio web https://bl.ocks.org/ tiene una amplia galería de ejemplos gratuitos, modulares y funcionales. Estos ejemplos se pueden expandir para su uso particular en el proyecto.
    b. GraphViz puede exportar imágenes estáticas SVG para uso web, pero no permite interacciones. La librería externa no oficial Canviz[20] facilita algunas interacciones limitadas.

[17] https://github.com/d3/d3/wiki/Requests#d3_json
[18] http://www.graphviz.org/doc/info/output.html
[19] http://js.cytoscape.org/#notation/elements-json
[20] https://github.com/koops/canviz - GitHub - koops/canviz: Fork of Canviz by Ryan Schmidt

c. Cytoscape.js: Las interacciones ya vienen programadas. No es posible expandirlas.

4) Posibilidad de visualizar resultados jerárquicamente
   a. D3.js: Gran variedad de visualizaciones jerárquicas.
   b. GraphViz: Solo visualización tradicional de grafos.
   c. Cytoscape.js: Solo visualización tradicional de grafos.

Se eligió D3.js, dado que tiene la posibilidad de leer sin modificaciones el formato JSON de resultados utilizado; se puede interactuar con sus visualizaciones, y ofrece una gran variedad de visualizaciones jerárquicas. No obstante, un aspecto importante a dejar constancia es que su curva de aprendizaje es mayor a las otras librerías. Prácticamente todas las interacciones y cálculos deben programarse, por lo que la complejidad de aprender el funcionamiento particular de la librería, funcionamiento SVG en HTML y javascript se suma.

Adicionalmente, en esta instancia se eligió el framework web para integrar la interfaz web con el algoritmo y así hacer la transición de aplicación de escritorio a aplicación web, y así cumplir con los objetivos 6 y 7 del TEG. Entre Django, Web2py y Flask, que fueron los framework web inicialmente considerados, se eligió Flask. Se eligió por el simple hecho de que éste framework es el que tiene menor curva de aprendizaje, se puede utilizar como si se estuviese programando un programa en Python, y no se tiene que aprender detalles y funcionamiento del framework en sí para utilizarlo. Es decir, las funcionalidades web que presenta están abstraídas y no es necesario saber ni de las librerías subyacentes, ni de una lógica de funcionamiento o flujo de trabajo particular del framework. Finalmente, la tecnología definitiva a utilizar en la tesis está descrita en la Tabla 16.

Tabla 16: Iteración 9 - Tecnología definitiva a usar

Lenguaje	Drivers de BD	Librerías de grafos	Framework web	Libería interfaz web
Python 2.7	Pymongo 2.8	NetworkX 1.1	Flask 0.10	D3.js v3.0.0

Fuente: Elaboración propia

Elección e implementación de la visualización

En el sitio web https://bl.ocks.org/ hay diferentes tipos de visualizaciones jerárquicas. Las opciones fueron: grafos tipo árboles, *treemaps* ("mapas de arboles"), círculos anidados, y *sunburst* ("rayos de sol"), tal como se ven en la Figura 9.

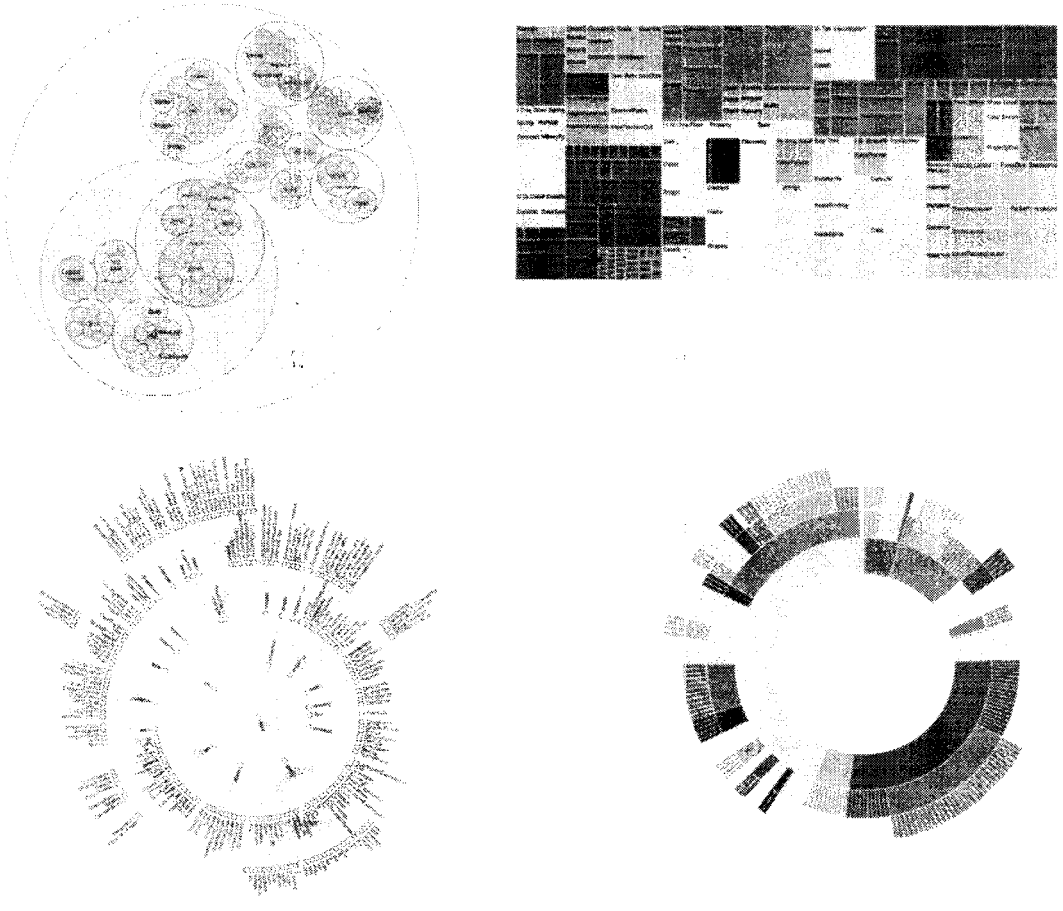

Figura 9: Visualizaciones candidatas

Leyenda: Sup. Izq: Circulos anidados, Sup. Der: treemap, Inf. Der: Sunburst, Inf. Izq: Grafo.
Fuente: Editado de **https://bl.ocks.org/**

Se eligió la visualización sunburst puesto que puede comunicar el significado de "general-particular" de cada obra. Es decir, la obra científica central, el

círculo más grande ya que es la obra introductoria más general, y los semi-círculos anidados van disminuyendo en tamaño según qué tan específica es una obra. Además, se puede modificar la visualización para mostrar los detalles de cada obra, y crear efectos de transición que complementan el sentido de profundización en el conocimiento.

Navegación y presentación de la información complementaria

En la interfaz web se mostrará la información básica sobre cada literatura científica que sea elegida. Por cada obra se presentará: título, palabras claves, número de citaciones, DOI, referencia donde fue publicado y su tipo de publicación.

Respecto a la navegación, siempre se mostrará una leyenda con los grupos y se identificarán con un color. Esto facilitará la elección del grupo que se desea leer al respecto. Además, cuando se seleccione una obra, la interfaz hará un efecto de acercamiento la obra seleccionada para ver sus detalles y las obras que están más próximas. Para facilitar el regreso al inicio de la interfaz, existe el botón "Vuelta a la raíz" para facilitar la elección de otro grupo/camino de búsqueda.

Los títulos de las obras se hicieron curvos y proporcionales al espacio que ocupan, para que la visualización sea más limpia y clara. Por limitantes de la forma de la interfaz existen casos en los que dependiendo de la longitud del título, la visualización del título será parcial. Por lo tanto, como apoyo visual, hay un cuadro de diálogo flotante que muestra el título completo cuando se posa el cursor encima de una obra. Sin embargo, ésta forma de presentar los títulos hace que la visualización se mucho más clara, dado que si se muestran los títulos horizontalmente la visualización pierde sentido y, además, los títulos se sobreponen entre sí.

El inicio de la interfaz está en la Figura 10, una vista general de los resultados en la, Figura 11, y los detalles de un resultado en la Figura 12.

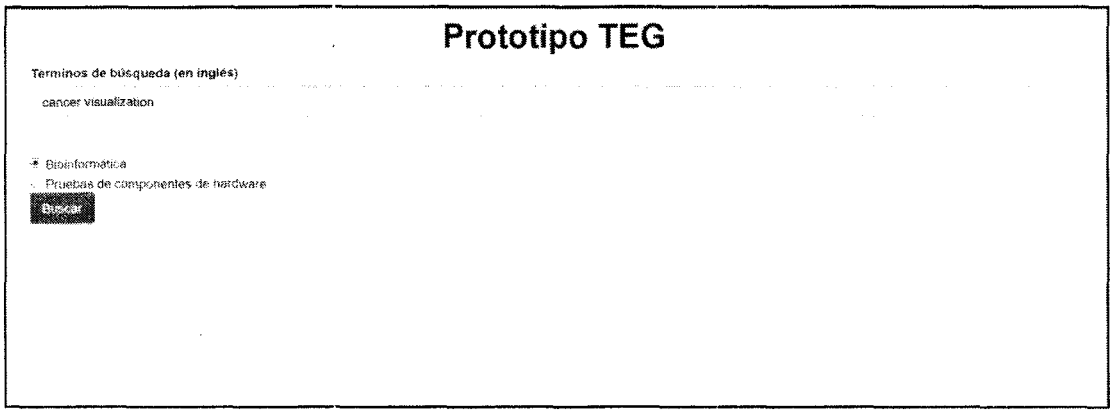

Figura 10: Interfaz – Inicio

Fuente: Elaboración propia

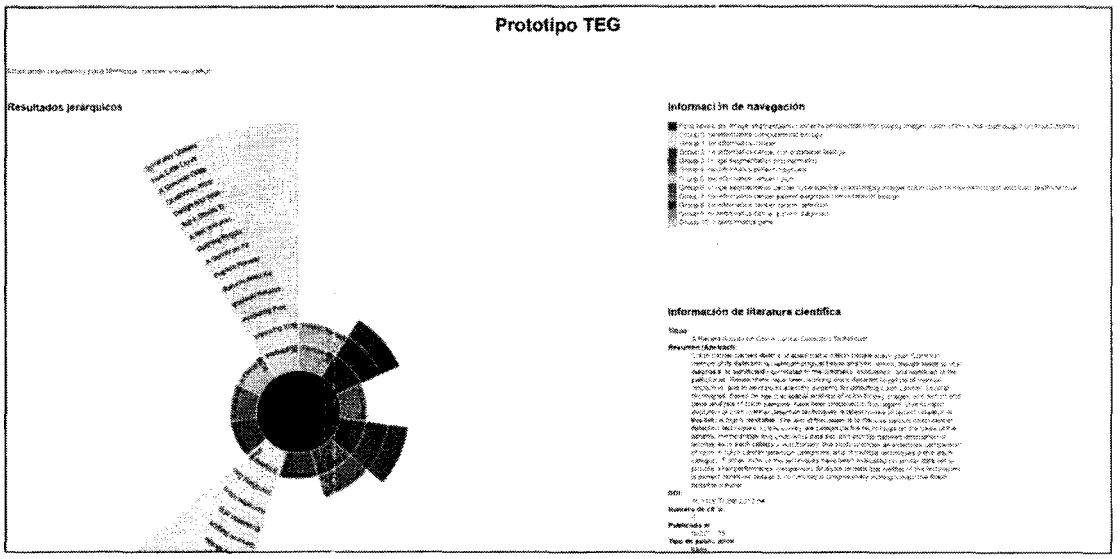

Figura 11: Interfaz – Resultados

Fuente: Elaboración propia

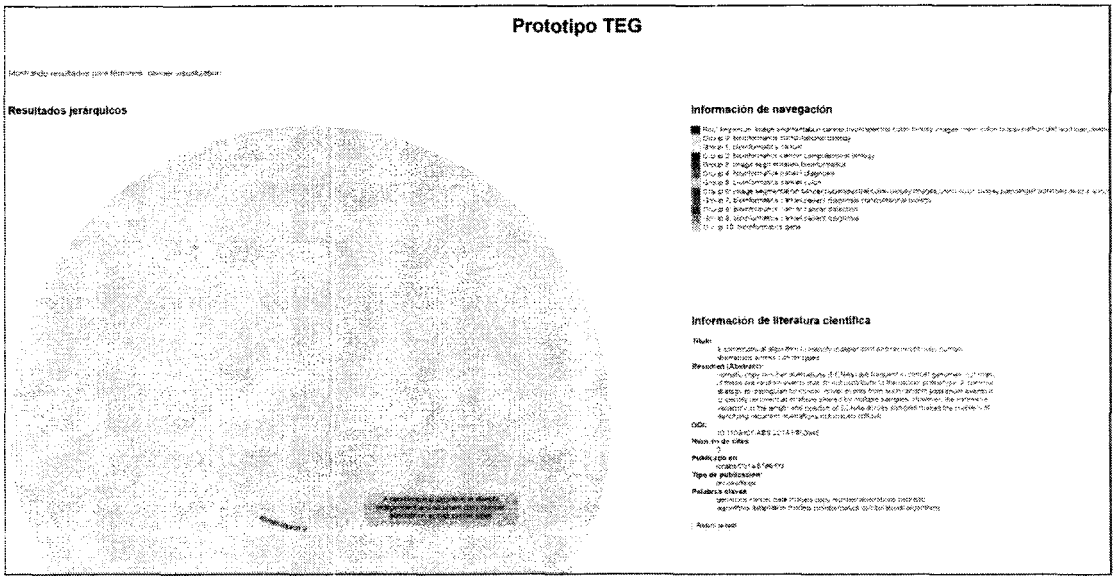

Figura 12: Interfaz - Detalle de resultados

Fuente: Elaboración propia

Una escenario de uso de la visualización es la siguiente: Aquel usuario que desee informarse sobre un el tema buscado en el área de conocimiento seleccionada, puede comenzar por la obra científica central, que normalmente será una obra de tipo secundario o terciario sobre el tema. Esto permite tener una idea general sobre el tema buscado. Luego, el usuario puede leer aquellas obras adyacentes de la literatura central, de los grupos que le interesen; esto es para ganar un mejor entendimiento del área, y luego decidir el grupo donde profundizar. Finalmente, luego de haber decidido el grupo o los grupos a profundizar, el usuario puede leer las obras en el orden mostrado, considerando que mientras más alejada del centro es una obra, más especializada es. Todas las obras tienen un color dependiendo del grupo a cual pertenecen. El usuario también está invitado a buscar con las palabras claves que aparecen en los grupos para hacer otras búsquedas que le puedan ser útil.

Esta interfaz también expone al usuario palabras claves relacionadas, y permite visualizar una gran cantidad de obras científicas ordenadas por grupo. Una forma que contrasta con solo ver de 10 a 20 resultados en una página, los cuales están ordenados por relevancia según los términos de búsqueda. Además, las palabras claves de los grupos están relacionadas por semánticamente relacionadas según la elección del autor.

Planificación

Ahora que se han cumplido e integrado todos los objetivos específicos del TEG, se procederá a realizar la evaluación de los resultados del algoritmo.

Décima iteración

Objetivos iteración

8) Realizar la evaluación de la eficacia del algoritmo de minería de datos

Riesgos y alternativas

Existe el riesgo de no encontrar expertos en las áreas ya modeladas. Ante esta situación se puede modelar un área de investigación donde se tenga disponibilidad de expertos y de datos en la base de datos

Desarrollo

Para evaluar la eficacia del algoritmo se hizo un cuestionario. El cuestionario fue creado junto con el tutor, y sus preguntas fueron diseñadas para evaluar la eficacia del algoritmo y para preguntar la opinión de la herramienta.

La lista entera de preguntas se encuentra en la sección ANEXOS, en la pág. 64. Las preguntas más importantes a continuación: Si los resultados son presentados en jerarquía; si la **Literatura Central Inicial (LCI)** y sus palabras claves, tienen relación con el área de investigación; si las ramificaciones tienen relación con el área de investigación y la **LCI**; y si literatura científica presentada en las ramificaciones siguen un orden de "general a particular"

respecto al contenido. Adicionalmente, se les preguntó sus opiniones sobre la herramienta en general y sus recomendaciones para ella.

Las áreas utilizadas en la evaluación del cuestionario fueron Bioinformática y Tolerancia a fallos en Hardware. Los expertos contactados a evaluar el sistema fueron el Prof. Stefano di Carlo y el Prof. Alfredo Benso, siendo ambos expertos en las áreas de Bioinformática y Tolerancia a fallos en Hardware[21] [22]. El cuestionario se les presentó a los expertos junto con la última versión de la herramienta.

Antes de responder a las preguntas del cuestionario, se les pidió a los expertos realizar al menos 3 búsquedas de palabras claves por área de investigación de su libre elección en cada área, para tener familiaridad con la herramienta y evaluar la herramienta bajo condiciones de uso real. De este modo se prueba la herramienta en al menos 12 casos, sobre los cuales se basará la opinión general. Las palabras claves buscadas por área de investigación se encuentran en la Tabla 17.

Tabla 17: Palabras claves buscadas por los evaluadores según área de investigación

	Areas de investigación	
Evaluadores	**Tolerancia a fallos en Hardware**	**Bioinformática**
Stefano Di Carlo	Software-based self-testing	miRNA
	March tests	protein function
	Memory testing	network simulation
	Cross-Layer reliability	gene regulation
	SBST	
Alfredo Benso	march test	Protein interaction
	stuck-at fault	protein function
	coupling fault	gene regulatory network

Fuente: Elaboración propia

[21] https://www.researchgate.net/profile/Stefano_Di_Carlo
[22] https://www.researchgate.net/profile/Alfredo_Benso

Las respuestas a cada pregunta por cada experto se encuentran en los anexos (ANEXOS, pág. 67). De las respuestas de los cuestionarios se destaca que ambos expertos coinciden en que:

- Los resultados son presentados jerárquicamente.
- Que usualmente la **Literatura Central Inicial (LCI)** escogida es un buen material para comenzar a leer sobre un tema y que sus palabras claves están relacionadas con el tema
- Que los artículos presentados en cada ramificación de la **LCI** (cada grupo) están ordenados de "general a particular".
- El tipo de visualización y la presentación de las obras científicas presentadas son útiles.

Respecto a las recomendaciones:

- El Prof. Benso resalta que quizás hay que dar más peso a las palabras claves que se introducen en la búsqueda para tener unos resultados más relevantes a las palabras introducidas.
- El Prof. Di Carlo resalta que no hay forma de saber qué tan 'cerca' o 'lejos' está una obra científica de otra.
- Ambos coinciden en que hace falta mostrar la lista de los autores en la información asociada a cada obra científica

Respecto a la opinión en general de la herramienta, El Prof. Di Carlo opina que en general esta herramienta 'sigue un camino nuevo e innovador', piensa que esta idea 'debería seguir siendo desarrollada'. El Prof. Benso opina que la herramienta es un 'muy buen resultado' y que demuestra que se puede conseguir 'más conocimiento de los datos'. Finalmente, ambos consideran que esta idea debería seguir siendo desarrollada.

A continuación se detallan los resultados obtenidos con respecto a los objetivos planteados. La Tabla 18 referencia las páginas y las iteraciones donde se desarrolló cada objetivo específico. Los otros resultados obtenidos también están detallados y referidos en páginas e iteración en la Tabla 18

Tabla 18: Referencias a páginas e iteraciones donde se desarrollaron objetivos del TEG

Objetivos	Páginas donde se desarrolló objetivo	Número de iteración
Objetivo específico 1	28 y 33	4,5
Objetivo específico 2	20	2
Objetivo específico 3	28 y 33	4,5
Objetivo específico 4	35 44	6,7
Objetivo específico 5	46	8
Objetivo específico 6	48	9
Objetivo específico 7	46 y 48	8, 9
Objetivo específico 8	55	10
Objetivo general	55	10
Otros resultados	48	9

Fuente: Elaboración propia

Objetivo específico 1 - Diseñar e implementar un modelo basado en grafos capaz de representar diferentes tipos de relaciones entre un conjunto de artículos científicos devueltos por una base de datos bibliográfica.

- La librería IEEE_GRAPH para manipulación de grafos.
- Modelo de grafo para mostrar las relaciones entre literatura científica palabras, claves y publicaciones científicas.

Objetivo específico 2 - Evaluar y seleccionar una base de datos de metadatos.

- Elección y descripción de base de datos IEEE CSDL, luego de evaluar bases de datos.

Objetivo específico 3 - Diseñar e implementar mecanismo de acceso de datos adecuado a los metadatos.

- Librería de mecanismo de acceso de datos (IEEE_DATA).

Objetivo específico 4 - Definir una métrica para evaluar el alcance y la relevancia de un artículo científico:

- Métrica propuesta para calcular alcance por palabra clave y obra científica.

Objetivo específico 5 - Diseñar e implementar un algoritmo de minería de datos capaz de analizar el grafo bibliográfico para identificar los nodos más relevantes (artículos científicos) y ordenarlos jerárquicamente por relevancia para un tema de investigación determinado.:

- Algoritmo para agrupar y ordenar artículos científicos según palabras claves semánticamente relacionadas por el/los autor(es).

Objetivo específico 6- Diseñar e implementar una interfaz de usuario web simplificada para ejecutar el algoritmo de minería de datos:

- Diseño, elección e implementación de interfaz

Objetivo específico 7 - Integrar el mecanismo de acceso de datos, el algoritmo de minería de datos, y la interfaz de usuario.:

- Integración de Interfaz, librerías de grafos, datos, y análisis de datos.

Objetivo específico 8 -Realizar la evaluación de la eficacia del algoritmo de minería de datos :

- Realización y análisis de encuestas para evaluar desempeño del algoritmo de minería de datos.

Otros resultados

- Lista de palabras claves semánticamente sugeridas.
- Selección de arquitectura de tecnología para realizar minería de datos basado en grafos. Python, Flask, Pandas, NetworkX, JSON, Excel, D3.JS.

CAPÍTULO VI. Conclusiones y Recomendaciones

Conclusiones

1) Se diseñó e implementó un algoritmo de minería de datos que proporciona un conjunto jerárquico de resultados que ayudan al usuario en la elección de un grupo de artículos científicos para leer en un tema de investigación.
2) El desempeño del algoritmo de minería de datos en las áreas de Bioinformática y Prueba de fallos de Hardware fue generalmente considerada como positiva por el panel de expertos. La interfaz, a su vez, fue unánimemente considerada positiva.
3) Los drivers de MongoDB con R no son software maduro (ver pág. 26, Tabla 8 Iteración 3 - Comparación de funcionalidad de drivers entre R y python). Por lo que no se aconseja usar R para aplicaciones web para usuarios, sino solo para hacer prototipos, pruebas e investigaciones.

Recomendaciones

1) Crear un tesauro usando la taxonomía de la ACM para asociar a las palabras claves, de las áreas de investigación, sinónimos y acrónimos.
2) De las áreas de conocimiento que se deriven, se podrían utilizar las palabras que la conforman para asignar automáticamente un(as) área(s) de conocimiento a una búsqueda de usuario.
3) Se recomienda mostrar una lista de candidatos a Literatura Científica Inicial (**LCI**), de la cual el usuario puede elegir el más relevante para

su búsqueda, y a partir de ahí construir la jerarquía, en vez de seleccionarlo automáticamente.

4) Se recomienda crear la jerarquía a partir del DOI de un artículo científico, para generar la jerarquía de resultados que tengan una métrica de ámbito mayor si se desea profundizar, o menor si se desea comprender las bases.
5) Añadir la posibilidad de graduar los resultados jerárquicos mediante la interfaz de usuario. Esto incluye, controlar número de ramificaciones, número de integrantes por rama, etc.

REFERENCIAS BIBLIOGRÁFICAS

NetworkX - Overview. (12 de 09 de 2015). Obtenido de NetworkX: https://networkx.readthedocs.io/en/stable/overview.html

Association, A. L. (03 de March de 2015). *Metadata.* Obtenido de American Library Association: http://www.ala.org/tools/atoz/metadata/metadata

Blake, G., & Bly, R. W. (1993). *The Elements of Technical Writing.* New York: Macmillan Publishers.

Boehm, B., & Hansen, W. J. (2000). *Spiral Development: Experience, Principles, and Refinements.* Pittsburg: Carnegie Mellon University.

Cardoso, L. (2006). *Sistemas de Bases de Datos II.* Caracas: Universidad Católica Andrés Bello.

Crockford, D. (25 de Junio de 2015). *Introducing JSON.* Obtenido de Introducing JSON: http://www.json.org/

Feather, J., & Sturges, P. (2003). *International Encyclopedia of Information and Library Science.* New York City: Routledge.

Foundation, P. S. (14 de 04 de 2015). *General Python FAQ¶.* Obtenido de General Python FAQ¶: https://docs.python.org/2/faq/general.html

Freitas, A. (2002). *Data Mining and Knowledge Discovery with Evolutionary Algorithms.* Secaucus: Springer-Verlag New York, Inc.

González Tous, M., & Mattar V, S. (2012). Las claves de las palabras clave en los artículos científicos. *Revista MVZ Córdoba*, 2955-2956.

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques.* San Francisco: Morgan Kaufmann.

Hlava, M. (07 de Marzo de 2012). *What is a Taxonomy?* Obtenido de KMWorld Magazine: http://www.kmworld.com/Articles/Editorial/What-Is-.../What-is-a-Taxonomy-81159.aspx

IEEE. (25 de July de 2015). *The Institute of Elecrical and Electronical Engineers (IEEE).* Obtenido de "Information for Authors" kit - IEEE: https://www.ieee.org/documents/info_authors_kit.pdf

Pandas, E. d. (25 de marzo de 2015). *pandas 0.16.0 documentation.* Obtenido de pandas: powerful Python data analysis toolkit: http://pandas.pydata.org/pandas-docs/version/0.16.0/

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Vanderplas, J. (2012). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 2825-2830.

Piatetsky, G. (18 de August de 2014). *Four main languages for Analytics, Data Mining, Data Science.* Obtenido de KDnuggets: http://www.kdnuggets.com/2014/08/four-main-languages-analytics-data-mining-data-science.html

Ronacher, A. (2015). *Flask Documentation 0.10.*

Samatova, N. F., William, H., Jenkins, J., Padmanabhan, K., & Chakraborty, A. (2014). *Practical graph mining with R.* Boca Raton: CRC Press.

Schembri, P. (13 de 12 de 2007). The different types of scientific literature. Msida, Malta.

Thayer, R., & Boehm, B. (1988). *Tutorial--software engineering project management.* Washington, D.C.: Computer Society Press of the IEEE.

The MongoDB 2.6 Manual. (24 de Julio de 2015). Obtenido de The MongoDB 2.6 Manual: https://docs.mongodb.com/v2.6/

University, I. (24 de 08 de 2015). *What is Metadata?* Obtenido de Indiana University Knowledge Base: https://kb.iu.edu/d/aopm

Venables, W. N., Smith, D. M., & Team, R. D. (2004). *An Introduction to R.*

Xiao-Li Li, C. S.-K. (2009). Searching for Rising Stars in Bibliography Networks. *Lecture Notes in Computer Science*, 288-292.

Yan, R., Huang, C., Tang, J., Zhang, Y., & Li, X. (2012). To Better Stand on the Shoulder of Giants. *JCDL.*

Extracto de clasificación de palabras claves

palabra_clave	part_suavizada(pc)
ieee transactions	0.892136853
robinson-foulds distance	0.647901963
evolution (biological)	0.647901963
molecular configurations	0.631634537
incomplete lineage sorting	0.620617824
computational biology	0.578352506
reverse-complement	0.533212371
labeled expression data	0.502006963
heuristic programming	0.502006963
deep coalescence	0.502006963
exact enumerative algorithm	0.46963854
closest substring problem	0.46963854
bioinformatics (genome or protein) databases	0.447369428
molecular biophysics	0.442846264
tree local similarity	0.436017044
coexpression	0.436017044
rna local similarity	0.436017044
differential coexpression	0.436017044
sequence-structure matching	0.436017044
structural bioinformatics	0.436017044
bioinformatics	0.422731068
biology and genetics	0.418705321
biological pathway knowledge	0.401041533
variation calling	0.401041533
chinese cabbage	0.401041533
gene translation	0.401041533
prior probability construction	0.401041533
extended burrows-wheeler transform	0.401041533
supertree	0.401041533
pedigrees	0.401041533
pan-genome indexing	0.401041533
conserved gene clusters	0.401041533
optimal bayesian classifier (obc)	0.364598357
protein-ligand binding sites	0.364598357
incomplete annotations	0.364598357
phenotype classification	0.364598357
np-hardness,satsolver, idp	0.364598357
clustering validity index	0.364598357
synthetic pathway generation	0.364598357
blood serum analysis	0.364598357
gene tree parsimony	0.364598357
string transformation	0.364598357
gene teams	0.364598357
protein complex prediction	0.364598357
genetic algorithms	[illegible]
parameter estimation	[illegible]
alignment	[illegible]
complexity theory	[illegible]
simulated annealing	[illegible]
bayesian inference	[illegible]
inference algorithms	0.009041781
software libraries	0.008896103
parallel processing	0.008865155
monte carlo methods	0.008862653
graphics processing unit	0.008852501
design methodology	0.008728293
regression analysis	0.008452875
probabilistic logic	0.008213835
python	0.008060192
upper bound	0.008001467
ensemble learning	0.008411601
filtering theory	0.008369[illegible]
junctions	0.008349531
data handling	0.008284489
markov processes	0.008285489
parallel algorithms	0.008237924
graphical models	0.008219744
unsupervised learning	0.008198457
linear programming	0.008189777
graphics processing units	0.008052769
entropy	0.007997148
predictive models	0.007944[illegible]
soft computing	0.007927817
regularization	0.007927817
engineering	0.007912134
encoding	0.007896099
arrays	0.007847218
strain	0.007806102
data structures	0.007769377

Extracto de clasificación de palabras claves (en español)

palabra_clave	part_suavizada(pc)
ieee transactions	0.89213692
distancia robinson-foulds	0.64790196
evolución (biológica)	0.64790196
configuraciones moleculares	0.63163454
ordenamiento de descendencia imcompleta	0.62061782
biología computacional	0.57835251
complemento reverso	0.53321237
datos de expresión etiquetados	0.50200696
programación heurística	0.50200696
Coalescencia profunda	0.50200696
Algoritmo enumerativo exacto	0.46963854
Problema de subcadena más cercano	0.46963854
Bases de datos de bioinformática (genoma o proteína)	0.44736943
biofísica molecular	0.44284626
similaridad de árbol local	0.43601704
coexpresión	0.43601704
similaridad local de rna	0.43601704
coexpresión diferencial	0.43601704
emparejamiento secuencia-estructura	0.43601704
bioinformática estructural	0.43601704
bioinformática	0.42273107
biología y genética	0.41870532
conocimiento de la vía biológica	0.40104153
llamada de variación	0.40104153
col china	0.40104153
traducción genética	0.40104153
onstrucción previa de probabilidad	0.40104153

Transformación extendida de burrows-wheeler	0.40104153
superárbol	0.40104153
pedigrees	0.40104153
Indexación pan-genoma	0.40104153
Clusters de genes conservados	0.40104153
Clasificador bayesiano óptimo	0.36459836
Sitios de unión de proteína-ligando	0.36459836
Anotaciones incompletas	0.36459836
Clasificación del fenotipo	0.36459836
np-hardness,satsolver, idp	0.36459836
índice de validez de agrupamiento	0.36459836
generación de vías sintéticas	0.36459836
análisis de transfusión de sangre	0.36459836
parsimonia de arbol genético	0.36459836
transformación de cadena de caracteres	0.36459836
grupos de genes	0.36459836
predicción compleja de proteínas	0.36459836
algoritmos genéticos	0.00922459
estimación de parámetros	0.00906019
alineación	0.00845928
teoría de la complejidad	0.00813928
recocido simulado	0.00806917
inferencia bayesiana	0.00803414
algoritmos de inferencia	0.00801878
librerías de software	0.00868710
procesamiento en paralelo	0.00868518
métodos de monte carlo	0.00862951

unidad de procesamiento gráfico	0.00882501
metodología de diseño	0.00872785
análisis de regresión	0.00871488
logica probabilística	0.00871459
python	0.00868015
límite superior	0.00860328
aprendizaje en grupo	0.00841160
teoría de filtrado	0.00839479
uniones	0.00834950
manejo de datos	0.00828848
procesos de markov	0.00828649
algoritmos paralelos	0.00823772
modelos gráficos	0.00821074
aprendizaje no supervisado	0.00820600
programación lineal	0.00818978
unidades de procesamiento gráfico	0.00807277
entropía	0.00797917
modelos predictivos	0.00795401
computación suave	0.00794[illegible]
regularización	0.00792626
ingeniería	0.00791213
condificación	0.00789610
arrays	0.00784722
raza	0.00780610
estructura de datos	0.00778613

Extracto de taxonomía IEEE CS

B Hardware
- **B.1** Control Structures and Microprogramming
  - B.1.3 Control Structure Reliability, Testing, and Fault-Tolerance
    - B.1.3.a Diagnostics
    - B.1.3.b Error-checking
    - B.1.3.c Redundant design
    - B.1.3.d Test generation
- **B.2** Arithmetic and Logic Structures
  - B.2.3 Reliability, Testing, and Fault-Tolerance
    - B.2.3.a Diagnostics
    - B.2.3.b Error-checking
    - B.2.3.c Redundant design
    - B.2.3.d Test generation
- **B.3** Memory Structures
  - B.3.4 Reliability, Testing, and Fault-Tolerance
    - B.3.4.a Diagnostics
    - B.3.4.b Error-checking
    - B.3.4.c Redundant design
    - B.3.4.d Test generation
- **B.4** I/O and Data Communications

D Software/Software Engineering
F Theory of Computation
H Information Technology and Systems
- **H.1** Models and Principles
- **H.2** Database Management
  - **H.2.8** Database Applications
    - **H.2.8.a** Bioinformatics (genome or protein) databases
- **H.3** Information Storage and Retrieval
- **H.4** Information Technology and Systems Applications
  - **H.4.1** Office Automation
    - **H.4.1.d** Spreadsheets
      - **H.4.1.e** Time management
      - **H.4.1.f** Word processing

K Computing Milieux
L. Haptics
M Services Computing
O Affective Computing

Thesis questionary

Name and surname: Alfredo Benso

Keywords searched in bioinformatics: Protein interaction (good), protein function (good), gene regulatory network (average)

Keywords searched in Hardware fault tolerance: march test, stuck-at fault, coupling fault

General instructions before answering:

Choose a topic in which you wish to research. Write keywords related to your research interests and the knowledge area selected as search input. Do please perform more than 2 searches, and then proceed to answer the questions.

Question set 1:

1) Are the results presented hierarchically? yes

2) Do you consider that the first scientific publication, the publication in the center,

a. Is related with the selected topic? Yes. It usually represents a good general paper or review on the argument. It is a good "starting paper" for learning a new topic

b. Has related keywords to the selected topic? Yes, but I noticed that sometimes the relationship is a bit loose.

3) Please observe the top right of the results site. Do you consider that ramifications (groups) are:

a. Related to the selected topic? Yes they do. Some of the categories are ok but some are a bit too general or not focused enough on the initial keywords. I am not sure this is something that can be solved with the current data quality of the database.

b. Related to the central scientific publication? Yes. Nevertheless, some of the papers are a bit far from the initial keywords and I have a feeling some papers are missing but I could not check further. The quality of the search results can be improved, but only after an improvement of the quality of the original data. Also, I agree with my colleague Dr Carlo in the idea of possibly using the CSDL original search result + this new visualization solution. But this is for another project ☺

4) First, select the central scientific publication, and read the associated information (on the lower right). Then choose a group or two that you are interested in. Read each article (and its associated information) of the chosen group(s). Do you consider:

a. That the articles are presented from general (more basic) to particular (more research specific)? Yes

b. Please give a brief opinion about the order of the articles. The papers are in "general to specific" order. I did not read all the papers of my searches, but from the title some seem loosely related to the central paper.

Question set 2:

Please perform another search.

1) Do you consider this type of visualization useful? YES

2) Do you consider this type of visualization more user friendly compared to a plain list of results? Some important info are missing, like the Authors' list, but it is a VERY INTERESTING IDEA

3) Do you consider this type of tool useful for your research. Yes, but more for somebody that wants to learn a new topic

4) Please give your opinion, in general, about the visualization and the tool. The type of visualization is exactly what I was looking for. For sure there is more work to be done on the graphic, but this was not the goal of the thesis. In my opinion (and the opinion of my fellow colleagues in the Computer Society) this is a very good result and prototype that shows how we can get more knowledge from the available data.

Thesis questionary

Name and surname: Stefano Di Carlo

Keywords searched in Bioinformatics: Software-based self-testing (very poor), March tests (poor), Memory testing (average), Cross-Layer reliability (poor), SBST (gives error)

Keywords searched in Hardware fault tolerance: miRNA (poor), protein function (good), network simulation (average), gene regulation (average)

General instructions before answering:
Choose a topic in which you wish to research. Write keywords related to your research interests and the knowledge area selected as search input. Do please perform more than 2 searches, and then proceed to answer the questions.

Question set 1:

5) Are the results presented hierarchically? YES

6) Do you consider that the first scientific publication, the publication in the center:

a. Is related with the selected topic? PARTIALLY, but usually the central paper is general enough and quite well chosen

b. Has related keywords to the selected topic?
YES, but sometimes the relationship with the selected topics is VERY general

7) Please observe the top right of the results site. Do you consider that ramifications (groups) are:

a. Related to the selected topic? Partially yes. It seems that they are more general than they should be. Also it seems keywords are considered as single words. How can you search for a multi-word keyword?

b. Related to the central scientific publication? Whereas the central publication seem ON topic, many of the other papers seem related to the original topic but many times in a loose way.

8) First, select the central scientific publication, and read the associated information (on the lower right). Then choose a group or two that you are interested in. Read each article (and its associated information) of the chosen group(s). Do you consider:

a. That the articles are presented from general (more basic) to particular (more research specific)? YES

b. Please give a brief opinion about the order of the articles.
It is clear that articles are sorted from general to specific, but the "closeness" of each paper to the original searched keywords is not always clear.

Question set 2:
Please perform another search.

5) Do you consider this type of visualization useful? ABSOLUTELY YES

6) Do you consider this type of visualization more user friendly compared to a plain list of results? (Yes/No)
Some important info are missing, like the Authors' list, but it is a VERY INTERESTING IDEA

7) Do you consider this type of tool useful for your research? Could be

8) Please give your opinion, in general, about the visualization and the tool.

It seems that the searched keywords usually DO NOT appear in the title of the search results. Maybe it is better to give more weight to it? In general, the tool goes in a new and innovative direction. I never saw results displayed in such a hierarchical way.

Could it be possible to give a "score" to each paper so that the user can understand immediately how much a paper is close to the original keywords?

In general I think this idea should be developed further, but it might need more training to capture all the key papers from keywords. Why not to start from the list of papers that the traditional CSDL search gives, and then try to sort those out in hierarchical way?