Datos no Estructurados: Indexación, Búsquedas y Aplicaciones

Darío Ruano, Paola Azar, Andrea Maldonado , Norma Herrera

Laboratorio de Investigación y Desarrollo en Bases de Datos

Universidad Nacional de San Luis, San Luis, Argentina

dmruano@email.unsl.edu.ar, epazar@unsl.edu.ar, andreamaldonadoma@gmail.com, nherrera@unsl.edu.ar

Andrés Pascal

Departamento Ingeniería en Sistemas de Información

FRCU, Universidad Tecnológica Nacional, Entre Ríos, Argentina

andrespascal22@gmail.com

Resumen

Los avances en las tecnologías de la información y la comunicación permitieron que las bases de datos crezcan de manera exponencial en tamaño y se diversifiquen en contenido. Un gran porcentaje de los datos disponibles son no estructurados, por lo que los mismos no son asimilables con los modelos clásicos de bases de datos, como el modelo relacional. Por esta razón, se necesita contar con nuevos modelos de datos y nuevas herramientas para el manejo de los mismos. En este proyecto nuestro interés está centrado al estudio de modelos para bases de datos no estructutados y técnicas de indexación y búsqueda asociados a los mismos.

Contexto

El presente trabajo se desarrolla en el ámbito de la línea Técnicas de Indexación para Datos no Estructurados del Proyecto Tecnologías Avanzadas de Bases de Datos (22/F814), cuyo objetivo es realizar investigación sobre manejo y recuperación eficiente de información no tradicional que implique el manejo de datos no estructurados. Este proyecto forma parte de Laboratorio de Investigación y Desarrollo en Bases de Datos (LaBDa) de la Universidad Nacional de San Luis.

1 Introducción

El crecimiento exponencial de las fuentes de información disponibles, ha provocado que las bases de datos crezcan en volumen y en tipo de datos almacenados. En la actualidad, gran parte de la información disponible involucra el uso de datos no estructurados (sonidos, imágenes, video, huellas digitales, etc) que no admiten la representación clásica de bases de datos relacionales, donde la información se organiza como nuplas (filas) en relaciones (tablas). Los procesos de búsqueda se enfrentan a nuevos desafíos relacionado al rendimiento y al tipo de respuesta esperada.

Es en este contexto donde surgieron nuevos modelos de bases de datos capaces de cubrir las necesidades de las aplicaciones que manejan datos no estructurados [3]. Entre esos

modelos encontramos el modelo de *bases de datos de texto*, el *modelo de espacios métricos* y el modelo *métrico-temporal.*

En el caso de **bases de datos de texto**, la base de datos consiste de una gran colección de texto sobre la que se requiere resolver consultas de manera eficiente. Esta colección de texto se conceptualiza como un única y gran secuencia de caracteres T, que se encuentra dividida en varios archivos. Este texto T puede representar no sólo lenguaje natural, sino también música, códigos de programas, secuencias de ADN, secuencias de proteínas, etc. Una de las búsquedas más comunes en bases de datos de texto es la *búsqueda de un patrón*: el usuario ingresa un string P (*patrón de búsqueda*) y el sistema retorna todas las posiciones del texto T donde P ocurre. Resolver la búsqueda de un patrón de manera eficiente en grandes colecciones de texto requerirá la construcción de un índice.

El modelo de **espacios métricos** [4, 9, 12] permite modelar y manejar datos no estructurados sobre los que se quieren realizar búsquedas por similitud. El espacio está formado por un conjunto de objetos $\mathcal{X}$ y una función de distancia d definida entre ellos que mide cuan similares son. La base de datos es cualquier subconjunto finito $\mathcal{U} \subseteq \mathcal{X}$. Una de las consultas más comunes en este modelo de bases de datos es la *búsqueda por rango*. En esta búsqueda dado un elemento $q \in \mathcal{X}$, al que llamaremos *query*, y un radio de tolerancia r, la búsqueda por rango consiste en recuperar los objetos de la base de datos cuya distancia a q no sea mayor que r.

El modelo **métrico-temporal** [10, 2] fue pensado para manejar objetos no estructurados con tiempos de vigencia asociados, en los que existe necesidad de consultar por similitud y por tiempo en forma simultánea. Un *espacio métrico-temporal* es un par (U, d), donde $U = O \times N \times N$, y la función d es de la forma $d : O \times O \rightarrow R^+$. Cada elemento $u \in U$ es una triupla (obj, t_i, t_f), donde obj es un objeto (por ejemplo, una imagen, sonido, cadena, etc) y $[t_i, t_f]$ es el intervalo de vigencia de obj. La función de distancia d, que mide la similitud entre dos objetos, cumple con las propiedades de una métrica (positividad, simetría y desigualdad triangular). Una *consulta métrico-temporal* implica buscar todos los objetos o de la base de datos que estén a una distancia a lo más r de q, y cuyo tiempo asociado t se solape con el tiempo de la consulta.

En todos estos modelos, para resolver eficientemente las consultas planteadas, se necesitan índices que sean específicos a cada tipo de dato y tipo de consulta planteada.

En este proyecto nos dedicamos al estudio, diseño y optimización de algoritmos de indexación y búsquedas sobre datos no estructurados y exploramos dominios de aplicación de las técnicas diseñadas. La aplicación de los métodos y técnicas diseñadas en casos reales de estudio nos permiten realizar transferencia tecnológica al medio y medir el desempeño de nuestras propuestas en problemas reales de la comunidad. En la actualidad nos encontramos trabajando sobre problemas relacionados a la medicina forense.

2 Líneas de Investigación

Describimos a continuación las principales líneas en las que nos encontramos trabajando actualmente.

2.1 Indexación en Memoria Secundaria

El proceso de paginación de un índice consiste en dividir el mismo en partes, cada una de las cuales se aloja en una página de disco. Luego el proceso de búsqueda consiste en ir cargando en memoria principal una parte, realizar la búsqueda en memoria principal sobre esa parte, para luego cargar la siguiente y proseguir la búsqueda.

Cuando un índice se maneja en disco, el costo de búsqueda queda determinado por la cantidad de accesos a disco realizadas [11]. Aun así, es importante no descuidar las operaciones que se hacen en memoria principal a fin de lograr un funcionamiento eficiente del índice.

En la actualidad nos encontramos trabajando en la implementación de dos índices para memoria secundaria: el l *Trie de Sufijos (TS)* [8, 6] para bases de datos de texto y el *Historical-FHQT*(H-FHQT) [7] para bases de datos métrico-temporales.

En ambos casos hemos diseñado una técnica de paginado basándonos en el algoritmo de paginado para árboles binarios presentado en [5]. Esta técnica consiste en particionar el árbol en componentes conexas, denominadas *partes*, cada una de las cuales debe tener un tamaño que no supere la capacidad de una página de disco. El algoritmo procede en forma bottom-up tratando de condensar en una única parte un nodo con uno o más subárboles que dependen de él. En este proceso de particionado las decisiones se toman en base a la profundidad de cada nodo involucrado, donde la profundidad indica la cantidad de accesos a disco que deberá realizar el proceso de búsqueda para llegar desde esa parte a una hoja del árbol. En el caso del H-FHQT, se tiene en cuenta además de la profundidad, otros parámetros propios del diseño del índice para decidir qué partes condensar.

En este sentido, se ha avanzado en la implementación del TS y del H-FHQT en memoria secundaria, encontrándonos en la etapa de testing de las mismas.

2.2 Espacios Métricos y Medicina Forense

En medicina legal y forense existen varios temas de interés que necesitan manejar datos no tradicionales para resolver problemas de manera eficiente. Uno de esos problemas es la identificación de cadáveres NN en el contexto de búsqueda de personas desparecidas.

Dentro de los individuos que ingresan a los distintos Institutos de Medicina Forense del país, existen casos que no poseen las condiciones adecuadas para su identificación inmediata (indocumentados, en avanzado estado de descomposición, restos óseos, etc.) o sin posibilidad de identificación (fragmentos muy pequeños, restos carbonizados, etc.). Frente a esto, las instituciones deben investigar no sólo para determinar qué fue lo que sucedió (causa de la muerte) y cuándo sucedió (data de la muerte), sino también para poder dar con la identidad del cuerpo.

Claramente la identificación de cadáveres está directamente relacionada con la búsqueda de personas desaparecidas. En Argentina, no existe un sistema único de procesamiento para esta problemática. Cada provincia tiene su gobierno, su sistema forense y sus protocolos. Esto dificulta el proceso de identificación de cadáveres: si una persona desaparece en Chaco y aparece un cadáver similar en Chubut, no hay una forma rápida y correcta de relacionarlos. El Sistema Federal de Búsqueda de Personas Desaparecidas y Extraviadas (SIFEBU) es un intento de crear esta base de datos unificada pero sin tener automatizado el proceso de búsqueda

En esta línea abordamos la aplicación de la teoría de Espacios Métricos para la identificación de cadáveres en el contexto de búsqueda de personas desaparecidas. El objetivo es desarrollar un sistema que permita mantener una base de datos, modelizada con un espacio métrico, con información sobre cadáveres no identificados para posteriormente realizar búsquedas de personas desaparecidas.

Para ello, por cada cadáver mantenemos un vector con los datos de las características físicas del mismo. Al momento de realizar una búsqueda, se deberá ingresar las características físicas de la persona buscada y con esa información generar el vector característico de la persona buscada. A partir de este vector se re-

aliza una búsqueda por similitud sobre la base de datos de cadáveres, obteniendo como resultado una lista de cadáveres con características similares a la persona buscada, rankeados según el grado de similitud. Hasta el momento, el trabajo desarrollado es el siguiente:

Generación de Vectores Característicos Para la elaboración del sistema, como primer paso hubo que definir un core de datos que seas adecuado a la problemática de identificación. Usar pocos datos podría provocar que las búsquedas que se realicen sean de muy baja selectividad y usar demasiados puede provocar que se descarten elementos de la base de datos que sean de interés. En este sentido, ya hemos definido un primer core de datos sobre el cual trabajar: *color de ojos, color de pelo, color de piel, existencia de tatuajes y lugar de los mismos, cicatrices y/o marcas* (lunares por ejemplo), si existen *amputaciones y en qué lugar del cuerpo*, la *contextura física (atlético, atrófico, etc.)* y finalmente la existencia de *agenesias*. Para cada dato, se transforman los valores de su dominio en números que reflejen el grado de similitud entre los valores considerados y se genera el vector correspondiente. Claramente no todos los datos tienen el mismo grado de importancia, por ejemplo el color de pelo es menos importante que el color de ojos porque una persona puede cambiarse el color de pelo pero no el de ojos. Esto hay que tenerlo en cuenta para establecer para cada dato un peso que corresponda con el grado de importancia del mismo.

Función de Distancia Otro punto importante es la función de distancia a utilizar en las búsquedas. En una primera etapa usaremos la función coseno [1]. Si q es la query (vector de la persona buscada) y d_j es el *j-ésimo* vector de la base de datos, entonces el grado de similitud entre los vectores $\vec{d_j}$ y $\vec{q}$ se calcula como el coseno del ángulo formado entre ambos vectores.

Indexación y Búsquedas Con respecto al algoritmo de indexación comenzaremos usando algoritmos basados en pivotes. Cuando la base de datos se cargue con datos reales, se podrá analizar la dimensionalidad del espacio métrico sobre el cual se está trabajando y de ser necesario se cambiará el algoritmo de indexación. Con respecto a las búsquedas, utilizaremos las búsquedas de los k vecinos más cercanos, porque es la que más se adecúa a este problema. Esto permitirá al usuario del sistema decidir cuántos elementos desea recuperar en una primera instancia y luego, de ser necesario, podrá ampliar la búsqueda; por ejemplo: puede pedir los 10 cadáveres mas parecidos a la persona buscada y posteriormente puede ampliar la búsqueda pidiendo los 10 siguientes.

En función de los resultados que se obtengan con la primer versión del sistema se realizarán, de ser necesario, cambios que pueden implicar: aumentar o disminuir los datos del core, cambiar la función de distancia y/o cambiar el algoritmo de indexación.

Una extensión inmediata de esto, es mantener una bases de datos con los vectores característicos de las personas desaparecidas para que cada vez que ingresa un cadaver NN a un Instituto Forense del país, se pueda realilzar la correspondiente búsqueda sobre la base de datos de personas desaparecida.

3 Resultados Esperados

Como resultados de los trabajos que estamos realizando, esperamos obtener índices eficientes en memoria secundaria para las bases de datos planteadas. Todo el trabajo realizado nos permite adquirir la experiencia suficiente como para abocarnos al diseño de otros índices que sean competitivos en este ámbito.

En cuanto a la transferencia tecnológica, el resultado esperado es un sistema web de

identificación de cadáver en el contexto de búsqueda de personas desaparecidas.

4 Formación Recursos Humanos

Dentro de esta línea se forman docentes y alumnos de la Universidad Nacional de San Luis y de la Universidad Tecnológica Nacional (FRCU). Actualmente hay en desarrollo 2 Tesis de Maestría y un Trabajo Final de la Licenciatura, todos de la Universidad Nacional de San Luis.

Bibliografia

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[2] A. De Battista, A. Pascal, N. Herrera, and G. Gutierrez. Metric-temporal access methods. *Journal of Computer Science & Technology,*, 10(2):54–60, 2010.

[3] H.M. Blanken, A.P. de Vries, H.E. Blok, and L. Feng. *Multimedia Retrieval*. Data-Centric Systems and Applications. Springer Berlin Heidelberg, 2007.

[4] E. Chávez, G. Navarro, R. Baeza-Yates, and J.L. Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, September 2001.

[5] D. Clark and I. Munro. Efficient suffix tree on secondary storage. In *Proc. 7th ACM-SIAM Symposium on Discrete Algorithms*, pages 383–391, 1996.

[6] J. Cornejo, D. Ruano, and N. Herrera. Una mejora en tiempo del trie de sufijos. In *Congreso Argentino de Ciencias de la Computación*, Rio Cuarto, Córdoba, Argentina, 2019.

[7] A. De Battista, A. Pascal, G. Gutierrez, and N. Herrera. Un nuevo índice métrico-temporal: el historical-fhqt. In *Actas del XIII Congreso Argentino de Ciencias de la Computación*, Corrientes, Agentina, 2007.

[8] G. H. Gonnet, R. Baeza-Yates, and T. Snider. *New indices for text: PAT trees and PAT arrays*. Prentice Hall, New Jersey, 1992.

[9] Filip Nalepa, Michal Batko, and Pavel Zezula. Enhancing similarity search throughput by dynamic query reordering. In Sven Hartmann and Hui Ma, editors, *Database and Expert Systems Applications*, pages 185–200, Cham, 2016. Springer International Publishing.

[10] A. Pascal, A. De Battista, G. Gutierrez, and N. Herrera. Métodos de acceso para bases de datos métrico - temporales. In *Actas del XV Congreso Argentino de Ciencias de la Computación*, pages 1061–1070, Jujuy, Argentina, 2009.

[11] Jeffrey Scott Vitter. *Algorithms and Data Structures for External Memory*. Publishers Inc, 2006.

[12] Pavel Zezula. Similarity management of data: The DISA experience. In Massimo Mecella, Giuseppe Amato, and Claudio Gennaro, editors, *Proceedings of the 27th Italian Symposium on Advanced Database Systems, Castiglione della Pescaia (Grosseto), Italy, June 16-19, 2019*, volume 2400 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.