UNIVERSIDAD NACIONAL DE SAN LUIS
FACULTAD DE CIENCIAS FÍSICO
MATEMÁTICAS Y NATURALES

Trabajo Final de la Licenciatura en Ciencias de la Computación

Una Aplicación de Búsquedas por Similitud en el Ámbito del Comercio Electrónico

Libertad Speranza
María de los Ángeles de la Torre

Director: M.Cs. Norma Edith Herrera

San Luis - Argentina, Marzo 2020

Resumen

Sección Vacía

Índice general

Capítulo 1

Introducción

1.1. Introducción

Las bases de datos tradicionales son construidas basándose en el concepto de búsqueda exacta: la base de datos es dividida en registros y cada registro contiene campos completamente comparables. Las consultas a la base de datos retornan todos aquellos registros cuyos campos coinciden exactamente con los aportados en tiempo de búsqueda (búsqueda exacta).

A través del tiempo las bases de datos han evolucionado para incluir la capacidad de almacenar nuevos tipos de datos tales como imágenes, sonido, video, etc. Estructurar este tipo de datos en registros para adecuarlos al concepto tradicional de búsqueda exacta es difícil en muchos casos y hasta imposible si la base de datos cambia más rápido de lo que se puede estructurar (como por ejemplo la web). Aún cuando pudiera hacerse, las consultas que se pueden satisfacer con la tecnología tradicional están limitadas en variaciones de la búsqueda exacta.

Nos interesan las búsquedas en donde se puedan recuperar objetos similares a uno dado. Este tipo de búsqueda se conoce con el nombre de búsqueda por similitud, y surge en diversas áreas; tales como reconocimiento de voz, reconocimiento de imágenes, compresión de texto, biología computacional, entre otras.

Todas estas aplicaciones tienen algunas características comunes. Existe un universo X de objetos y una función de distancia $d : X \times X \to \Re$ que modela la similitud entre los objetos. El par (X, d) es llamado espacio métrico REF. La base de datos es un conjunto $U \subseteq X$, el cual se preprocesa a fin de resolver búsquedas por similitud eficientemente.

Una de las consultas típicas en este nuevo modelo que implica recuperar objetos similares de una base de datos es la **búsqueda por rango**, que denotaremos

con $(q, r)_d$. Dado un elemento de consulta q, al que llamaremos query y un radio de tolerancia r, una búsqueda por rango consiste en recuperar aquellos objetos de la base de datos cuya distancia a q no sea mayor que r.

Otra consulta que se utiliza para recuperar objetos similares es la **búsqueda de los k-vecinos más cercanos**, donde tenemos el elemento de consulta q y la cantidad de objetos de base de datos que se quieren recuperar, que llamaremos k. La cercanía de estos k objetos está dada por el valor de la función de distancia d hasta la query q.

Los cómputos realizados durante una búsqueda implican cálculos de la función de distancia d, operaciones adicionales (sumas, restas, comparaciones, etc) y tiempo de I/O si el índice y/o los datos se encuentran en memoria secundaria. Las búsquedas por similitud pueden ser resueltas trivialmente por medio de una búsqueda exhaustiva, con una complejidad $O(n)$. Para evitar esta situación, se preprocesa la base de datos por medio de un algoritmo de indexación con el objetivo de construir una estructura de datos o índice, diseñada para ahorrar cálculos en el momento de resolver una búsqueda.

Básicamente existen dos enfoques para el diseño de algoritmos de indexación en espacios métricos: uno está basado en Diagramas de Voronoi [REF] y el otro está basado en pivotes [REF]. En este trabajo abordamos el estudio de algoritmos de indexación basados en pivotes, enfocándonos en una aplicación que utilice este tipo de algoritmos en el ámbito del comercio electrónico.

Los algoritmos basados en pivotes construyen el índice basándose en la distancia de los objetos de la base de datos a un conjunto de elementos preseleccionados llamados pivotes.

1.2. Objetivo

El comercio electrónico, también conocido como e-commerce consiste en la compra y venta de productos o de servicios a través de la web. El desarrollo de nuevas tecnologías ha permitido que la capacidad y volumen de las comunicaciones se expanda de una manera exponencial, esto ha facilitado que el comercio electrónico tenga también un crecimiento exponencial.

Para el desarrollo de un sitio de comercio electrónico hay varios problemas que deben resolverse tales como administración de categorías de productos, búsqueda de productos, encriptación de datos, registración de usuarios, administración de medios de pago, entre otros. En este trabajo nos hemos centrado específicamente en el problema de búsqueda de productos.

Nuestra meta es utilizar búsquedas por similitud sobre las descripciones asociadas a los productos con el fin de estudiar el desempeño de los algoritmos basados en pivotes en un caso real.

1.3. Como se organiza la tesis

Hemos organizado este informe en dos partes: En la primera parte se introducen los conceptos necesarios para comprender el informe y en la segunda parte presentamos el trabajo realizado y los resultados que obtuvimos al aplicar los algoritmos de indexación.

Primera Parte

Capítulo 1: Definimos la motivación y los objetivos del trabajo.

Capítulo 2: Presentamos los conceptos básicos de las búsquedas en espacios métricos con el objetivo de dar el marco teórico necesario para comprender y fundamentar el desarrollo realizado en este trabajo final. También explicamos la situación actual del comercio electrónico, sobre el cual vamos a aplicar los conceptos teóricos.

Capítulo 3: Definimos completamente en qué consiste la aplicación de espacios métricos a esta base de datos real de comercio electrónico.

Segunda Parte

Capítulo 4: Describimos el trabajo realizado. Presentamos la implementación de dos algoritmos de indexación basados en pivotes: Selección de pivotes en forma random y selección de pivotes en forma incremental. También detallamos las diversas variaciones implementadas a fin de encontrar la que mejor se adapte al tipo de búsqueda que queremos realizar.

Capítulo 5: Mostramos la evaluación experimental de las variaciones implementadas.

Capítulo 6: Presentamos las conclusiones obtenidas como así también algunos puntos interesantes para abordar en futuros trabajos.

Capítulo 2

Conceptos Previos

Este capítulo introduce el marco teórico del presente trabajo a través de la definición de los siguientes conceptos: espacios métricos (ejemplo: diccionario de palabras), medidas de distancia; se incluyen algunas de las medidas más comunes utilizadas dentro de la temática a tratar, e introduciremos un modelo formal para las búsquedas por similitud.

2.1. Espacios Métricos

En este trabajo se aborda el tema de búsqueda no convencional, es decir, dentro de un universo de datos, nos interesa encontrar aquellos objetos que son "similares.ª un objeto dado. El modelo formal que abarca este tipo de búsquedas se denomina Espacio Métrico.

Espacio Métrico: Sea X un universo de objetos válidos y sea $U \subseteq X$ un subconjunto finito de tamaño n sobre el que se realizarán búsquedas. Llamarémos U a las base de datos, diccionario o simplemente conjunto de elementos. Definimos una función $d : (X \times X) \rightarrow \Re$ que denotará una medida de distancia entre objetos de X, esto significa que a menor distancia más cercanos o similares son los objetos. Esta función d cumple con las propiedades características de una función de distancia:

(a) $\forall x, y \in X d(x, y) \geq 0$ (Positividad)

(b) $\forall x, y \in X d(x, y) = d(y, x)$ (Simetría)

(c) $\forall x \in X d(x, x) = 0$ (Reflexividad)

en la mayoría de los casos:

(d) $\forall x, y \in X x \neq y \Rightarrow > 0$ (Positividad estricta)

Estas propiedades sólo aseguran una definición consistente de la función, pero no pueden usarse para evitar comparaciones durante una búsqueda por similitud. Para que la función d sea realmente una métrica debe satisfacer la siguiente propiedad:

(e) $\forall x, y, z \in X d(x, y) \leq d(x, z) + d(y, z)$ (Desigualdad triangular)

El par X, d es llamado espacio métrico. Si la función d no satisface la propiedad de positividad estricta (d), entonces diremos que el espacio es pseudo métrico. Estos casos pueden ser fácilmente resueltos; basta con adaptar las técnicas de espacios métricos de manera tal que todos los objetos a distancia cero se identifiquen como un único objeto. Esto funciona dado que $d(x, x) = 0 \Rightarrow \forall z, d(x, z) = d(y, z)$ (puede demostrarse utilizando la desigualdad triangular). En algunos casos podemos tener cuasi métricas donde no se cumpla la propiedad de simetría (b). En estos casos podemos definir a partir de la función d una nueva función d', que sí sea métrica $d'(x, y) = d(x, y) + d(y, x)$.

Finalmente podemos llegar a relajar la desigualdad triangular (e) permitiendo $d(x, y) \leq \alpha d(x, z) + \beta d(z, y) + \gamma$. En estos casos podemos seguir usando los mismos algoritmos de espacio métrico, previo escalamiento.

2.1.1. Ejemplos de Espacios Métricos

Diccionario de Palabras

En este caso, los objetos del espacio métrico son cadenas de caracteres de un determinado lenguaje. Para medir la distancia entre cadenas, una función posible es la conocida como distancia de edición o distancia de Levenshtein. La distancia entre dos palabras o cadenas se define como el mínimo número de operaciones atómicas (insert, delete y replace) necesarias para transformar una palabra x en una palabra y. Las operaciones se definen formalmente de la siguiente forma:

- **insert**: inserción del caracter c en la cadena x en la posición i:
  $ins(x, i, c) = x_1 x_2 .... x_i c x_{i+1} ... x_n$
- **delete**: eliminación del caracter c en la posición i de la cadena x:
  $del(x, i) = x_1 x_2 .... x_{i-1} x_{i+1} ... x_n$
- **replace**: sustitución de un caracter c de la cadena x en la posición i por un nuevo caracter c':
  $repl(x, i, c) = x_1 x_2 .... x_{i-1} c' x_{i+1} ... x_n$

Claramente, esta es un función de distancia discreta y tiene una variedad de aplicaciones en recuperación de texto, procesamiento de señales y biología computacional. En este trabajo utilizamos algo similar al diccionario de palabras, ya que nuestro universo se compone de títulos de productos en venta publicados en un sitio de e-commerce.

Espacios vectoriales

Los espacios vectoriales k-dimensionales son un caso especial de espacios métricos. Hay varias funciones de distancia para espacios vectoriales, pero las más usadas son las pertenecientes a la familia Ls o familia de distancias de Minkowsky, definidas como:

$$L_s((x_1, ..., x_k), (y_1, ..., y_k)) = (\sum_{1=i}^{k} |x_i - y_i|^s)^{1/s}, 1 \leq s < \infty \tag{2.1}$$

Para el caso de $s = \infty$, se toma el límite de la fórmula anterior para s tendiendo a. Se obtiene como resultado, que la distancia entre dos puntos del espacio vectorial es la máxima diferencia entre sus coordenadas:

$$L_\infty((x_1, ..., x_k), (y_1, ..., y_k)) = max_{1 \leq i \leq k} |x_i - y_i| \tag{2.2}$$

La siguiente figura ejemplifica la búsqueda por rango $(q, r)_d$ sobre un conjunto de puntos en $\Re^2$, usando como función de distancia L_2 (izquierda) y L_∞ (derecha).

IMAGEN

Figura 2.1: Ejemplos de búsqueda por rango (q,r)d, con d=L2 (izquierda) y con d=L (derecha)

Las líneas punteadas representan aquellos puntos que están exáctamente a distancia r de q. En consecuencia, todos aquellos elementos que caen dentro de estas líneas forman parte del resultado de la búsqueda. La distancia L_2, más conocida como *Distancia Euclidiana*, se corresponde con nuestra noción de distancia espacial. Por otro lado, las búsquedas con distancia Lse corresponde con la búsqueda por rango clásica, donde el rango es un hiper-rectángulo k dimensional.

En el ámbito de espacios vectoriales existen soluciones eficientes para búsquedas por similitud, como por ejemplo: $KD-tree$, $R-tree$, $Quadtree$ y $X-tree$ entre otras. Estas técnicas usan información sobre coordenadas para clasificar y agrupar puntos en el espacio. Desafortunadamente, las técnicas existentes son afectadas por la dimensión del espacio vectorial. Por lo tanto, la complejidad de búsqueda por rango depende exponencialmente de la dimensión del espacio.

Los espacios vectoriales pueden presentar grandes diferencias entre su dimensión representacional y su dimensión intrínseca (es decir la dimensión real en la que se pueden embeber los puntos manteniendo la distancia entre ellos). Por ejemplo, un plano embebido en un espacio de dimensión 50, tiene una dimensión

representacional de 50 y una dimensión intrínseca de 2.

Por esta razón, muchas veces se recurre a espacios métricos generales, aun sabiendo que el problema de búsqueda es más difícil. No se debe descartar que también es posible tratar un espacio vectorial como un espacio métrico general usando solo la distancia entre los puntos. Una ventaja inmediata de esto es que toma peso la dimensión intrínseca del espacio, independientemente de cualquier dimensión representacional.

2.2. Medidas de Distancia

Cuando hablamos de medidas de distancia, podemos dividirlas en dos grupos, de acuerdo al tipo de valores que retornan:

- **Discretas**: son aquellas que retornan solo un pequeño conjunto de valores (predefinido).
- **Continuas**: son las que retornan valores de un conjunto cuya cardinalidad es muy grande o infinita.

Un ejemplo de función continua es la distancia Euclideana entre vectores; mientras que la función de edición entre cadenas de caracteres (utilizada en los experimentos de este trabajo) representa una función discreta.

A continuación presentamos distintos ejemplos de funciones de distancia.

Distancia de Minkowski

Estas funciones forman realmente una familia de funciones denominadas métricas L_p, ya que los casos individuales dependen del parámetro numérico p. Estas funciones se definen sobre vectores n-dimensionales de números reales como:

$$L_p[(x_1, ..., x_n), (y_1, ..., y_n)] = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p} \tag{2.3}$$

Donde la métrica L_1 es conocida como la distancia de Manhattan, la métrica L_2 denota la distancia Euclidiana y la métrica $L = max_{i=1}^{n}|x_i - y_i|$ se denomina la distancia máxima, infinita o distancia de tablero de ajedrez.

Estas funciones son utilizadas en varios casos donde los vectores numéricos tienen condenadas independientes, por ejemplo, en mediciones de experimentos

científicos, observaciones ambientales, o el estudio de diferentes aspectos de procesos de negocios.

Distancia de la forma cuadrática

Esta distancia es utilizada en ámbitos en los que existe una correlación entre las dimensiones que componen el vector, ya que tiene el poder de modelar este tipo de dependencias. Un ejemplo de universo de aplicación de esta función son las bases de datos de imágenes.

En las funciones de la forma cuadrática, la medida de distancia entre dos vectores n-dimensionales está basada en una matriz positiva de $n \times nM = [m_{i,j}]$ donde los pesos $m_{i,j}$ denotan cuán fuerte es la conexión entre los componentes i y j de los vectores $\bar{x}$ e $\bar{y}$, respectivamente. Generalmente estos pesos son normalizados de manera que $0 \leq m_{i,j} \leq 1$ donde las diagonales $m_{i,i} = 1$. La siguiente expresión representa una medida de distancia cuadrática generalizada d_M, donde T denota la transposición del vector.

$$d_M(\bar{x}, \bar{y}) = \sqrt{(\bar{x} - \bar{y})^T \cdot M \cdot (\bar{x} - \bar{y})} \tag{2.4}$$

El cálculo de ésta distancia de la forma cuadrática puede ser muy caro en términos computacionales, dependiendo de la dimensionalidad de los vectores.

Distancia de Edición

La cercanía entre secuencias de símbolos (cadenas) puede medirse de manera efectiva a través de la distancia de Edición, también conocida como distancia de Levenshtein. La distancia entre dos cadenas $x = x_1...x_n$ e $y = y_1...y_n$ está definida como el número mínimo de operaciones atómicas de edición (inserción, eliminación y reemplazo) necesarias para transformar la cadena x en la cadena y. En términos formales, las operaciones de edición se definen como sigue:

- *inserción*: insertar el caracter c dentro de la cadena x en la posición i:
  $ins(x, i, c) = x_1 x_2 ... x_i c x_{i+1} ... x_n$
- *inserción*: eliminar el caracter en la posición i de la cadena x:
  $elim(x, i) = x_1 x_2 ... x_{i-1} x_{i+1} ... x_n$
- *reemplazo*: reemplazar el caracter en la posición i en x con el nuevo caracter c:
  $reemplazar(x, i, c) = x_1 x_2 ... x_{i-1} c x_{i+1} ... x_n$

La función de distancia de edición generalizada asigna pesos (números reales positivos) a cada operación atómica, por esto, la distancia entre las cadenas x e

y es el mínimo valor de la suma de los pesos de las operaciones atómicas necesarias para transformar x en y. Si los pesos asignados a cada operación difieren, la distancia de edición no es simétrica (violando la propiedad (b) del punto 2.1) y por lo tanto, no es una función métrica.

A los fines del presente trabajo, a todas las operaciones se le asigna un peso equivalente de uno.

Distancia de Edición de árbol

Esta distancia es una bien conocida medida de proximidad entre árboles, en la cual se define la distancia entre dos estructuras de árboles como el costo mínimo necesario para convertir el árbol fuente en el árbol destino utilizando un conjunto predefinido de operaciones de edición sobre árboles, tales como la inserción y la eliminación de un nodo. El costo individual de estas operaciones de edición puede ser constante para todo el árbol, o puede variar de acuerdo al nivel en el cual se lleva a cabo la operación. La razón de esta variación es que el costo de la inserción de un nodo cercano al nivel de la raíz puede ser más significativo que el costo de agregar un nodo hoja. Por supuesto, esto depende del dominio de aplicación.

Dado que los documentos XML generalmente se modelan como árboles etiquetados, esta distancia puede utilizarse también para medir las diferencias estructurales entre dos documentos XML.

Coeficiente de Jaccard

Si quisiéramos medir la distancia en un tipo diferente de datos, como lo son los conjuntos, podemos utilizar el denominado coeficiente de Jaccard. Asumiendo dos conjuntos A y B, el coeficiente de Jaccard se define como:

$$d(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|} \tag{2.5}$$

Esta función se basa simplemente en la razón entre las cardinalidades de la intersección y la unión de los conjuntos comparados. Como un ejemplo de una aplicación que trabaja con conjuntos, supongamos que tenemos un registro con usuarios y direcciones de sitios web que visitó cada uno. Para evaluar la similitud en el comportamiento de cada usuario, podemos utilizar el coeficiente de Jaccard.

2.3. Búsquedas por Similitud

Como ya mencionamos anteriormente, en este trabajo nos interesa realizar búsquedas por similitud; esto es, dada una consulta q, queremos encontrar todos

aquellos objetos del universo de interés que están cercanos a q, bajo una función de distancia dada. Básicamente, existen tres tipos de búsquedas por similitud:

1. Búsquedas por rango.
2. Búsquedas del vecino más cercano.
3. Búsquedas de los k vecinos más cercanos.

Las cuales explicaremos en mayor profundidad a continuación

2.3.1. Búsquedas por rango

Las búsquedas por rango $R(q, r)$ son probablemente las más comunes dentro de los espacios métricos. En estas búsquedas, la consulta se define como un objeto $q \in D$, especificando un radio r como una restricción de distancia. Dicha búsqueda devuelve todos los objetos encontrados a una distancia r de q, formalmente:

$$R(q, r) = o \in X, d(o, q) \leq r \tag{2.6}$$

Los objetos pertenecientes a la respuesta pueden ordenarse en base a su distancia de q, si es necesario. Nótese que el objeto q no necesita existir en el conjunto $X \subseteq D$ en el cual se busca, y la única restricción sobre q es que pertenezca al dominio D.

En la figura 1.2a se muestra un ejemplo de búsqueda por rango.

2.3.2. Búsquedas del vecino más cercano

Cuando se quiere realizar búsquedas por similitud sobre objetos utilizando búsquedas por rango, se debe especificar la distancia máxima para que los objetos sean incluídos en la respuesta; pero en ocasiones puede ser difícil especificar el radio sin conocimiento sobre los datos y la función de distancia. Por ejemplo, $r = 3$ de la métrica de distancia de edición representa menos de cuatro operaciones de edición entre dos cadenas comparadas; ésto tiene un claro significado semántico. Sin embargo, una distancia entre dos vectores de histograma de colores de imágenes es un número real cuya cuantificación no puede ser interpretada tan fácilmente.

Existe entonces, la situación en donde, si especificamos un radio de búsqueda muy pequeño no obtendremos ningún resultado, y deberemos repetir la búsqueda con un radio mayor. Por otro lado, si especificamos un radio de búsqueda muy grande, corremos el riesgo de que el costo computacional de calcular la distancia

sea muy alto y de que el conjunto de resultados contenga objetos no significativos.

Una alternativa para solucionar este tipo de problemas en la búsqueda por similitud es realizar una búsqueda de los vecinos más cercanos. Básicamente dicha búsqueda consiste en encontrar el objeto más cercano a la consulta q, al que denominamos vecino más cercano de q.

2.3.3. Búsquedas de los k vecinos más cercanos

El concepto de vecino más cercano puede generalizarse al caso de búsqueda de los k vecinos más cercanos. Específicamente $kNN(q)$ retorna los k vecinos más cercanos del objeto q, y dicho conjunto resultado se define formalmente como:

$$kNN(q) = R \subseteq X, |R| = k \wedge \forall x \in R, y \in X - R : d(q, x) \leq d(q, y) \qquad (2.7)$$

Cuando existen varios objetos a la misma distancia del k-ésimo vecino más cercano, la elección de uno se realiza arbitrariamente. La figura 2.2b ilustra la situación para una consulta $3NN(q)$; en ella los objetos o_1, o_3 están ambos a distancia 3,3 y el objeto o_1 es elegido como tercer vecino más cercano en forma aleatoria en vez de o_3.

IMAGEN

Figura 2.2: (a) Búsqueda por rango R(q,r) y (b) búsqueda del vecino más cercano 3NN(q).

2.4. Algoritmos de Indexación

Los algoritmos de indexación sirven para organizar la información o universo de datos en estructuras de datos o índices. Este pre proceso de la base de datos tiene como objetivo reducir el número de cálculos al momento de resolver una búsqueda. Un diseño eficiente de los *algoritmos de indexación*, junto con la desigualdad triangular, permite descartar elementos evitando comparar la query con cada uno de los elementos de la base o universo de datos. Así, dada una query, primero se recorre el índice para determinar un subconjunto de elementos candidatos a ser parte de la respuesta y luego se examinan esos conjuntos en forma exhaustiva para obtener la respuesta definitiva.

2.4.1. Un Modelo Unificado

Todos los algoritmos de indexación particionan la base de datos U en subconjuntos U_I. El índice permite identificar una lista de U_i que potencialmente contiene elementos significativos para la consulta. Por consiguiente, la búsqueda se reduce a:

- Recorrer el índice para obtener los candidatos. El costo de este proceso se denomina *complejidad interna*
- Examinar exhaustivamente estos candidatos a fin de encontrar los elementos que realmente forman el resultado de la búsqueda. El costo de este proceso se denomina *complejidad externa.*

La siguiente figura resume este modelo
IMAGEN Figura 2.3: Modelo general de algoritmos de indexación

Para comprender totalmente el proceso de indexación se necesita entender adecuadamente los conceptos de relación de equivalencia y partición de conjuntos. Aquí, la relevancia de dichos conceptos se debe a la posibilidad de particionar un espacio métrico en clases de equivalencias y obtener así un nuevo espacio métrico derivado del conjunto cociente.

Toda partición $\pi(X) = \{\pi_1, \pi_2, ..., \pi_n\}$ de un conjunto X induce una relación de equivalencia (que denotamos con $\widetilde{)}$. Inversamente toda relación de equivalencia induce una partición:

$$\forall x, y \in X : x\ y \Leftrightarrow xestenlamismaparteque y \tag{2.8}$$

Las clases de equivalencia $[x]$ de esta relación se corresponden con las partes π_i de la partición π, es decir,

$$\pi(X) = X/ \tag{2.9}$$

Por consiguiente, los algoritmos de indexación existentes definen una relación de equivalencia sobre el espacio métrico X. Las clases de equivalencia de X en el conjunto cociente $\pi(X)$ pueden considerarse como elementos de un nuevo espacio métrico, bajo alguna función de distancia $D : \pi(X) \times \pi(X) \to \Re^+$.

Ahora definimos la función D_0 que será de utilidad para encontrar la función D

$$D_0 : \pi(X)(X) \to \Re^+ D_0([x], [y]) : mnimo_{x\in[x], y\in[y]}\{d(x, y)\} \tag{2.10}$$

D_0 representa la menor distancia entre un elemento de la clase $[x]$ y un elemento de la clase $[y]$. Esta distancia es el máximo valor posible que mantiene el mapeo contractivo, es decir que $\forall x, y \in X : D_0([x],[y]) \leq d(x,y)$.

D_0 no puede ser utilizada con la finalidad de indexación porque no satisface la desigualdad triangular pero nos aproxima a una solución dado que sabemos que cualquier función de distancia D, que además cumple con las propiedades para ser una métrica, es una cota inferior de D_0 (manteniendo así el mapeo contractivo) que nos sirve para definir un nuevo espacio métrico. En otras palabras, esta nueva función D debe cumplir que $\forall [x],[y] en \pi(X), D([x],[y]) \leq D_0([x],[y])$.

Luego, podemos convertir un problema de búsqueda en otro, que esperamos sea más sencillo. Para una búsqueda $(q,r)_d$ primero, buscamos espacio $\pi(X,D)$ obteniendo como resultado $([q],r)_D = \{u \in U/D([u],[q]) \leq r\}$. Como D_0 es contractiva, podemos asegurar que $(q,r)_d \subseteq ([q],r)_D$. Luego realizamos una búsqueda exhaustiva en $([q],r)_D$ a fin de determinar los elementos que forman parte de la respuesta a la consulta $(q,r)_d$.

Hay dos enfoques generales para crear estas relaciones de equivalencia: uno está basado en pivotes y el otro se basa en particiones compactas o tipo Voronoi. Ambos se describen a continuación.

2.4.2. Algoritmos Basados en Pivotes

Los algoritmos de pivotes definen una relación de equivalencia basados en la distancia de los elementos a un conjunto de elementos preseleccionados que llamaremos *pivotes*.

Sea $\{p_1, p_2, ..., p_k\}$ un conjunto de pivotes, dos elementos son equivalentes si y sólo si están a la misma distancia de todos los pivotes:

$$x \; p_i y \Leftrightarrow d(x,p_i) = d(y,p_i) \forall_{i=1....k} \tag{2.11}$$

Se puede ver que cada clase de equivalencia está definida por la intersección de varias capas de esferas centradas en los puntos p_i como se muestra en la figura 2.4.2

IMAGEN Figura 2.4: Relación de equivalencia inducida por la intersección de anillos centrados en dos pivotes u8 y u11 (izquierda). Y la transformación de la query en un espacio vectorial de 2 dimensiones (derecha).

Buscaremos ahora una función de distancia D para la clase de equivalencia. Por la desigualdad triangular, para cualquier $x \in X$, se cumple que:

- 1. $d(p,x) \leq d(p,y) + d(y,x) \Leftrightarrow d(p,x) - d(p,y) \leq d(y,x)$
- 2. $d(p,y) \leq d(p,x) + d(x,y) \Leftrightarrow d(p,y) - d(p,x) \leq d(x,y)$
- 3. $d(p,x) - d(p,y) \leq d(y,x) \wedge d(p,y) - d(p,x) \leq d(x,y) \Leftrightarrow |d(x,p) - d(y,p)| \leq d(x,y)$

Esto es, para cualquier elemento p la distancia $d(x,y)$ no puede ser menor que $|d(x,p) - d(y,p)|$. Luego $D([x],[y]) = |d(x,p) - d(y,p)|$ es un límite inferior seguro para D_0. Si extendemos D para k los pivotes:

$$D([x],[y]) = max_{1 \leq i \leq k}\{|d(x,p_i) - d(y,p_i)|\} \tag{2.12}$$

Esta función de distancia D limita inferiormente a d y en consecuencia puede usarse como distancia en el espacio cociente.

La relación de equivalencia definida por un conjunto de k pivotes también puede considerarse como una proyección al espacio vectorial $\Re^k$. La i-ésima coordenada de un elemento es la distancia al i-ésimo pivote. Luego, a cada elemento x del espacio le corresponde el vector $\delta(x) = (d(x,p_1), d(x,p_2), ..., d(x,p_k)) \in \Re^k$. Entonces, dada un consulta de búsqueda $(q,r)_d$ debemos encontrar un conjunto de elementos candidatos $([q],r)_D = x : D([q],[x]) \leq r$. En este caso particular significa encontrar los elementos tales que:

$$max_{1 \leq i \leq k}\{|d(x,p_i) - d(q,p_i)|\} = L_\infty(\delta(x), \delta(q)) \leq r \tag{2.13}$$

Esto significa que hemos proyectado el espacio métrico original (X,d) en el espacio vectorial $\Re^k$ con la función de distancia L_∞. La figura 2.4.2 ilustra estas ideas para un conjunto de puntos usando sólo dos pivotes.

2.4.3. Algoritmos Basados en Particiones Compactas

La idea en este caso es dividir el espacio en particiones o zonas lo más compactas posibles, almacenando puntos representativos de dichas zonas, denominados centros, y algunos datos extra que permitan descartar zonas completas lo más rápidamente posible al momento de realizarse una consulta. Cada zona, a su vez, puede ser recursivamente particionada en más zonas, lo cual induce una jerarquía de búsqueda.

Criterio de partición de Voronoi

Los Diagramas de Voronoi [Aur91] han sido usados para búsquedas por proximidad en espacios vectoriales. Estos diagramas son una de las estructuras fundamentales dentro de la Geometría Computacional, de alguna forma ellos almacenan

toda la información referente a la proximidad entre puntos.

Se elige un conjunto de m centros $c_1, c_2, ..., c_m$. El resto de los objetos se asignan a la zona de su centro más cercano. Cuando todos los objetos de la base de datos son centros, el concepto de partición de Voronoi descrito coincide con el concepto de dominio de Dirichlet [3]. La Figura 2.4.3 ilustra este concepto en un espacio vectorial de dimensión 2.

IMAGEN Figura 2.5: Partición de Voronoi de un espacio vectorial de dimensión 2.

Al momento de realizar una consulta $(q, r)_d$, se evalúan las distancias entre q y los m centros, eligiendo el centro más cercano a q, el cual se denominará c.

Si la bola de la consulta $(q, r)_d$ intersecta la zona de algun centro distinto a c, entonces se debe revisar exhaustivamente esa zona para ver si hay objetos que caigan dentro de la bola de consulta (ver figura 2.4.4). Sea $x \in X$ un objeto perteneciente a la zona cuyo centro es c_i y que se encuentra a distancia menor o igual que r de la consulta q. Por la desigualdad triangular se tiene que:

1. $d(c, x) \leq d(c, q) + d(q, x) \leq d(c, q) + r$

2. $d(c_i, q) \leq d(c_i, x) + d(x, q) \leq d(c_i, x) + r \Rightarrow d(c_i, q) - r \leq d(c_i, x)$

Como x pertenece a la zona de c_i se cumple que $d(c_i, x) \leq d(c, x)$. De esta condición más (1) y (2) se obtiene:

3. $d(c_i, q) - r \leq d(c_i, x) \leq d(c, x) \leq d(c, q) + r \Rightarrow d(c_i, q) - r \leq d(c, q) + r$

Dada la condición (3), se pueden descartar las zonas cuyo centro c_i satisfaga la condición $d(c_i, q) > d(c, q) + 2r$, dado que en ese caso dicha zona no puede tener intersección con la bola de consulta

IMAGEN Figura 2.6: Condición de exclusión con criterio de partición de Voronoi.

2.5. Técnicas de Selección de Pivotes

La forma en que los pivotes son seleccionados puede afectar drásticamente la performance de un algoritmo. Una buena elección de pivotes puede en gran parte reducir los tiempos de búsqueda.

2.5.1. Criterios de eficiencia

Es necesario minimizar el número de evaluaciones de distancia que se realizan al momento de hacer una búsqueda por rango. Para lograr esto, los pivotes elegidos deben descartar la mayor cantidad de los elementos posibles antes de realizar una búsqueda dentro de una lista de elementos candidatos. En síntesis, un buen conjunto de pivotes debe generar una lista pequeña de elementos candidatos.

Sea (E, d) un espacio métrico. Un conjunto de pivotes $\{p_1, p_2, ..., p_k\} \in E$ definen un espacio P de tuplas de distancias entre pivotes y elementos del conjunto. Un elemento $x \in P$ se denotará como $[x]$ que es igual a la siguiente ecuación:

$$[x] = (d(x, p_1), d(x, p_2), ..., d(x, p_k)), x \in E, [x] \in P \tag{2.14}$$

Definimos la métrica $D = D_{\{p_1,...,p_k\}}$ del espacio P como:

$$D([q], [x]) = max_{i=1}^{k} |d(q, p_i) - d(x, p_i)| \tag{2.15}$$

Luego, obtenemos el espacio métrico (P, D) que resulta ser $(\Re^k, L_\infty)$. Dada una consulta por rango (q, r) es fácil ver este nuevo espacio métrico, que según la condición de la ecuación 2-6, no se pueden excluir aquellos elementos $x \in E$ que cumplen con:

$$D_{\{p_1,...,p_k\}}([q], [x]) \leq r \tag{2.16}$$

Para lograr que la cantidad de elementos elegido sea pequeña se debe maximizar la probabilidad de que $D_{\{p_1,...,p_k\}}([q], [x]) > r$. Una forma de lograr esto es maximizar la media de la distribución de distancias en P, la cual la llamaremos μ_p

2.5.2. Estimación del μ_p

La estimación del μ_p se estima como sigue:

- Elegir A pares de elementos $\{(a_1, a'_1), (a_2, a'_2), ..., (a_A, a'_A)\}$ del conjunto E, distintos entre sí.
- Mapear los A pares de elementos al espacio P y calcular la distancia D entre cada par de elementos, esto produce como resultado el conjunto finito de distancias $\{D_1, D_2, ..., D_A\}$.
- Luego de obtener las A distancias se estima el valor de p de la siguiente forma:

$$\mu_p = \frac{\sum_{i=1}^{A} D_i}{A} \tag{2.17}$$

De las ecuaciones 2.5.1 y 2.5.2, se puede deducir que dados un par de elementos (a, a') el costo de calcular $D([a], [a'])$ es $2k$ evaluaciones de la función d.

Se se utilizan A pares de elementos para estimar el valor de μ_p , se puede ver que el costo total de la estimación es de $2kA$ evaluaciones de la función d.

2.5.3. Métodos de selección de pivotes

En este trabajo vamos a describir los dos métodos de selección utilizados y sus costos en función al número de evaluaciones de la distancia d.

Selección Random

Esta técnica consiste en la elección al azar de los pivotes, no se usa ningún criterio de selección. En este trabajo mostraremos la diferencia o beneficios en las búsquedas al usar pivotes seleccionados usando técnicas incrementales versus la selección aleatoria de pivotes. El costo de optimizar los pivotes usando selección random es 0

Selección Incremental

El método consiste en elegir un pivote p_1 utilizando A pares de elementos del espacio E mapeados a P, tal que ése pivote maximice el μ_p. Luego elegir un segundo pivote p_2, tal que $\{p_1, p_2\}$ maximicen μ_p pero p_1 ya queda fijo. Luego elegir un tercer pivote p_3, tal que $\{p_1, p_2, p_3\}$ maximicen μ_p pero con p_1 y p_2 fijos. Repetir el proceso hasta elegir los k pivotes.

En cada iteración se elige un pivote de una muestra de tamaño X del espacio E, dado que buscar un elemento que maximice μ_p dentro del todo el conjunto E sería muy costoso.

Costo del algoritmo:

Si bien en cada iteración se estima el μ_p con los i pivotes seleccionados hasta el momento, no es necesario rehacer el cálculo completo si se almacena $D_{\{p_1,...,p_{i-1}\}}([a_r], [a'_r])\forall r \in 1..A$, es decir, para cada valor de $r = 1..A$ el valor máximo de $|d(a'_r, p_j) - d(a_r, p_j)|, j = 1..i-1$. En este caso solo se calcula la distancia con respecto a los pivotes candidatos $|d(a'_r, p_{cand}) - d(a_r, p_{cand})|, r = 1..A$ y se toma el valor máximo entre las dos distancias para el calculo de $D\{p_1, ..., p_i\}$. Entonces, si la muestra de donde se toman los pivotes candidatos es de tamaño X, en cada iteración se realizan $2AX$ evaluaciones de la función d. Luego, si se eligen k pivotes, el trabajo total realizado por el algoritmo tiene un costo de

$2kAX$ evaluaciones de la función d.

2.6. Dimensión en Espacios Métricos

El rendimiento de los algoritmo basados en pivotes empeora cuando la dimensión del espacio métrico aumenta. En el caso de los espacios vectoriales, su dimensión representacional es el número de coordenadas de los vectores que representan al conjunto; sin embargo, un espacio vectorial de alta representacionalidad puede poseer intrínsecamente una dimensionalidad baja.

2.6.1. Definición de dimensión intrínseca

La distribución de distancias de un espacio métrico se define como la probabilidad de que dos elementos se encuentre a cierta distancia l. Una aproximación a esta distribución es el *histograma de distancias*, el cual se construye desde un subconjunto del espacio métrico, calculando las distancias entre los elementos del subconjunto.

En [ref] se define la dimensión intrínseca de un espacio métrico como:

$$\gamma = \frac{\mu^2}{2\sigma^2} \tag{2.18}$$

Los parámetros μ y $\sigma 2$ son la media y la varianza, respectivamente, del histograma de distancias de los puntos que conforman dicho espacio métrico.

2.6.2. La maldición de la dimensionalidad

Cuando se realiza una consulta por rango, los algoritmos basados en pivotes intentan descartar la mayor cantidad de elementos del espacio métrico antes de realizar una búsqueda exhaustiva. Esto es, se genera una lista de elementos candidatos en la cual se verifica exhaustivamente la condición de búsqueda.

Si nos centramos nuevamente en el histograma de distancias de un espacio métrico (E, d), es fácil ver que según la condición de exclusión de elementos (ECUACION), dada una consulta (q, r) y un conjunto de k pivotes p_i, se pueden descartar todos los elementos $x \in E$ que no cumplan para algún $i \in 1..k$ la condición de la siguiente ecuación:

$$d(p_i, x) \notin [d(p_i, q) - r, d(p_i, q) + r] \tag{2.19}$$

Mientras más grande es la dimensión intrínseca del espacio métrico, la media del histograma aumenta y/o disminuye la varianza.

Cuando la varianza del histograma disminuye a medida que aumenta la dimensión del espacio, el número de elementos que se encuentran dentro del rango $[d(p,q) - r, d(p,q) + r]$ también aumenta, es decir que cada vez son menos los elementos que se pueden descartar. Alternativamente, si la media crece entonces r también crece con la dimensión del espacio si se quiere obtener un porcentaje fijo de elementos del conjunto. En espacios de alta dimensionalidad, casi todos los elementos se transforman en candidatos a la verificación exhaustiva, por lo que deben ser comparados directamente con la consulta. A esto se le denomina *maldición de la dimensionalidad*, y es independiente de la naturaleza del espacio métrico al cual pertenecen los elementos.

Capítulo 3

Una Aplicación de Espacios Métricos a Comercio Electrónico

Capítulo 4

Detalles de implementación

Capítulo 5

Evaluación experimental

En este capítulo presentamos los resultados de ejecutar el algoritmo de búsqueda por rango usando dos técnicas de selección de pivotes: random y incremental.

Comenzaremos describiendo el seteo experimental, los experimentos realizados y luego daremos el análisis de los resultados obtenidos.

En una primera etapa nos enfocamos en determinar el rango adecuado para las búsquedas y la forma en que íbamos a seleccionar los conjuntos de pivotes para cada una de las técnicas de selección.

5.1. Seteo Experimental

Para los experimentos usamos la base de datos de "*items*"(productos publicados) de Mercadolibre.

Agrupamos los productos en categorías que reúnen productos de similares características, teniendo como resultado 30 categorías distintas, que de ahora en adelante las llamaremos bases de datos.

Por cada base de datos y por cada técnica de selección, random e incremental, creamos índices y files de pivotes. Los números de pivotes elegidos fueron: 16, 32, 64, 128, 256, 512, 1024, 2048 y 4096. Los pivotes seleccionados fueron obtenidos por categorías. Para tomar esta decisión hicimos experimentos que más adelante mostraremos.

Luego, agrupamos las bases de datos en grupos segmentados según el procentaje que representan el número de pivotes sobre el tamaño de la base de datos, a este valor lo llamaremos ratio de pivotes. Esto es así porque tenemos bases de

datos que son muy chicas (ejemplo: 1034 items) y el ratio de pivotes era muy alto para esas categorías. De esta segmentación obtuvimos 4 grupos.

5.1.1. Selección del rango de búsqueda

La mayoría de los trabajos de investigación existentes sobre búsqueda por similitud en textos, utilizan como universo de datos diccionarios de palabras como por ejemplo inglés o español. Esto genera una base común que puede utilizarse a la hora de generar variaciones sobre algoritmos de búsqueda en nuevos trabajos, es decir, se comparte la base de datos y los parámetros de los experimentos, lo cual permite poder comparar fácilmente los resultados de los mismos.

Es común encontrar trabajos de búsqueda por rango donde la variación del radio es siempre entre 1 y 5 y la elección de ese valor no requiere más que una simple decisión azarosa por parte del autor.

En nuestro trabajo, el universo de datos es bastante más singular, ya que se trata de títulos de productos reales, cuya redacción está a cargo del usuario que publica el producto para su venta y donde la única limitante es el tamaño de ese título (60 caracteres).

Esta particularidad tiene como consecuencia un universo de datos variado y heterogéneo, donde cada elemento de dicho universo es una combinación de palabras, abreviaciones, números y caracteres especiales. Ante ésta combinación de características, el primer obstáculo que debimos sortear para comenzar con la evaluación experimental fue, precisamente, la selección del radio de búsqueda.

Como una primera aproximación, seleccionamos cuatro títulos de productos de la categoría con mayor cantidad de elementos y luego procedimos a realizar una búsqueda de los k-vecinos con:

- $k = 0{,}001 \times cantidad\_de\_elementos\_de\_la\_categoria$ (0,1 %)
- $k = 0{,}01 \times cantidad\_de\_elementos\_de\_la\_categoria$ (1 %)
- $k = 0{,}1 \times cantidad\_de\_elementos\_de\_la\_categoria$ (10 %)

Las búsquedas se realizaron utilizando la estrategia de selección de pivotes incremental, y la cantidad de pivotes elegida fue de 128.

Tamaño de base de datos	Número de k-vecinos		
	0.001 %	0.01 %	0.1 %
31.632	32	316	3163

Cuadro 5.1: Número de k-vecinos utilizados.

Títulos de búsqueda	Radio promedio		
	0.001 %	0.01 %	0.1 %
Libro Te Amo Pero Soy Feliz Sin Ti. Jaime Jaramillo	32	35	37
El Secreto Rhonda Byrne Lvbp13	21	24	26
Libro De Italiano Forza 2	14	17	23
Libro La Magia Rhonda Byrne El Secreto Lvbp13	28	32	35

Cuadro 5.2: Muestra para determinar el radio de búsqueda.

Títulos de búsqueda	Radio Promedio
Libro Te Amo Pero Soy Feliz Sin Ti. Jaime Jaramillo	32
El Secreto Rhonda Byrne Lvbp13	21
Libro De Italiano Forza 2	14
Libro La Magia Rhonda Byrne El Secreto Lvbp13	28
PROMEDIO GENERAL	**27**

Cuadro 5.3: Tabla de radio promedio.

Como podemos visualizar en los resultados, el promedio del radio de búsqueda arroja un valor de 27, valor que utilizamos para realizar algunas pocas búsquedas sobre el resto de las categorías. Al analizar los resultados obtenidos llegamos a la conclusión de que debíamos disminuír el valor del r, ya que en la mayoría de los casos la búsqueda retornaba una cantidad de elementos cercana a la totalidad de la base de datos y en otros casos los elementos retornados no eran similares al título de búsqueda. Ante éstos hallazgos, tomamos una segunda aproximación: primero obtuvimos el promedio de la longitud de los títulos para cada categoría, luego promediamos esos valores para obtener un único resultado y lo dividimos por 2. De esa forma llegamos al rango elegido para todos los experimentos: $r = 23$.

5.1.2. Selección de los conjuntos de pivotes

En este trabajo, además de las técnicas de selección de pivotes, random e incremental, consideramos dos políticas para la elección de los grupos de pivotes:

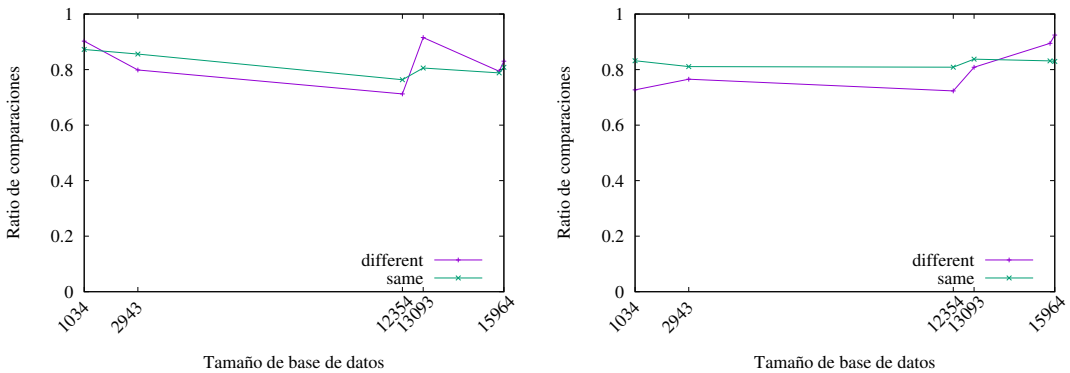
Mismo grupo de pivotes para todas las categorías o bases de datos: Esta estrategia, selecciona el grupo de pivotes considerando todas las bases de datos. Luego, el mismo grupo de pivotes, es utilizado en cada uno de los experimentos sobre las distintas bases de datos.

Diferentes grupos de pivotes por categoría o base de datos: Sobre cada una de las bases de datos se seleccionan los conjuntos de pivotes.

Para determinar cuál estrategia era la más adecuada, se seleccionaron conjuntos de pivotes, random e incremental, según las políticas arriba mencionadas para los grupos de 16 y 64 pivotes.

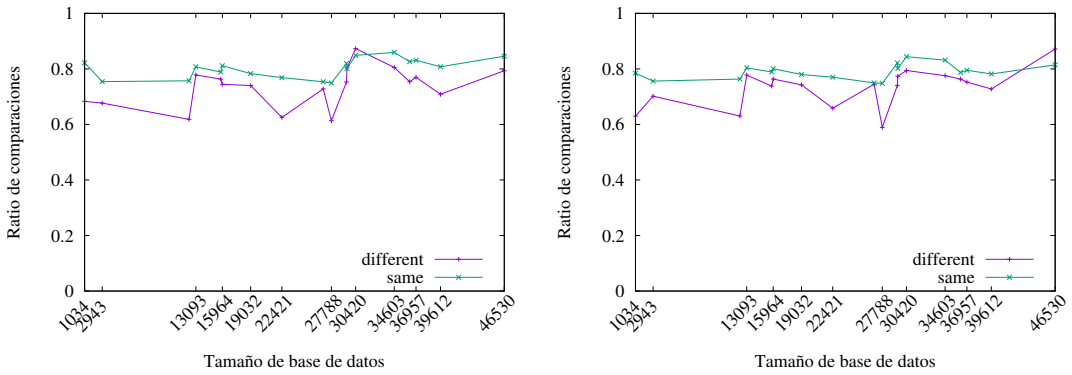
Por cada base de datos se eligió al azar el 10 % de los títulos como elementos de búsqueda. Para 16 pivotes, se usaron 6 bases de datos (tamaño total: 61.184) y se ejecutaron 12.232 búsquedas en total. Para 64 pivotes, se usaron 18 bases de datos (tamaño total: 440.631) y se ejecutaron 88.114 búsquedas en total.

Como se puede observar en la figura [**?**] A continuación, en las figura x.y.z y x.y.z', se muestra el ratio de comparaciones realizadas respecto del tamaño de la base de datos para 16 y 64 pivotes respectivamente.



(a) Selección incremental (b) Selección random

Figura 5.1: Estrategia de selección para 16 pivotes: mismos pivotes vs diferentes



(a) Selección incremental (b) Selección random

Figura 5.2: Estrategia de selección para 64 pivotes: mismos pivotes vs diferentes

Por lo que se puede observar, aunque la estrategia varía por cada base de datos, en términos generales se comporta mejor la política de selección *diferentes grupos pivotes por categoría o base de datos* para ambas técnicas; random e incremental.

5.2. Ejecución experimental

En una segunda etapa, nos concentramos en cómo agrupar los datos teniendo en cuenta la variación del tamaño de las bases de datos para luego poder analizar la información segmentada y poder sacar mejores conclusiones.

5.2.1. Creación de grupos

Dada la variabilidad de tamaños de las bases de datos, para poder analizar luego los resultados correctamente, definimos cuatro grupos segmentados por el tamaño de base de datos y por cantidad de pivotes. Esto es, calculamos el ratio de pivotes sobre el tamaño de las base de datos (DB), quedando la siguiente distribución:

En total se ejecutaron 276 experimentos. Esto es, 138 experimentos usando técnica de selección de pivotes random y 138 experimentos usando técnica de selección de pivotes incremental.

Se seleccionó al azar el 10 % de los elementos de cada una de las bases de datos para realizar las búsquedas. La cantidad total de búsquedas por rango realizadas fue: 339.899. Este valor incluye las búsquedas usadas para descartar la política

Grupo	Cantidad de DB	Rango del tamaño de DB	Número de Pivotes	Cantidad de Experimentos
1	6	[1.034 - 15.964]	16, 32, 64, 128, 256	60
2	12	[19.032 - 46.530]	64, 128, 256, 512, 1024	120
3	8	[57.198 - 13.6323]	256, 512, 1024, 2048	64
4	4	[16.7995- 21.3578]	512, 1024, 2048, 4096	32

Cuadro 5.4: Tabla de grupos.

de selección *Mismo grupo de pivotes para todas las categorías o bases de datos* mencionada anteriormente.

5.3. Análisis de resultados

Vamos a presentar los resultados segmentados en cuatro grupos y analizaremos tres enfoques diferentes.

Llamaremos ratio de comparaciones al porcentaje de comparaciones respecto del tamaño de la base de datos, esto es porcentaje de filtrado.

5.3.1. Efecto de las técnicas de selección de pivotes

En las figuras 5.5.x.y comparamos las técnicas de selección de pivotes, random e incremental, usando como métrica el número de evaluaciones de distancia para cada unos de los pivotes. De cada grupo mostraremos a continuación solo las base de datos de mayor tamaño; el resto se adjuntan en el anexo XX.

GRAFICAS POR GRUPOS

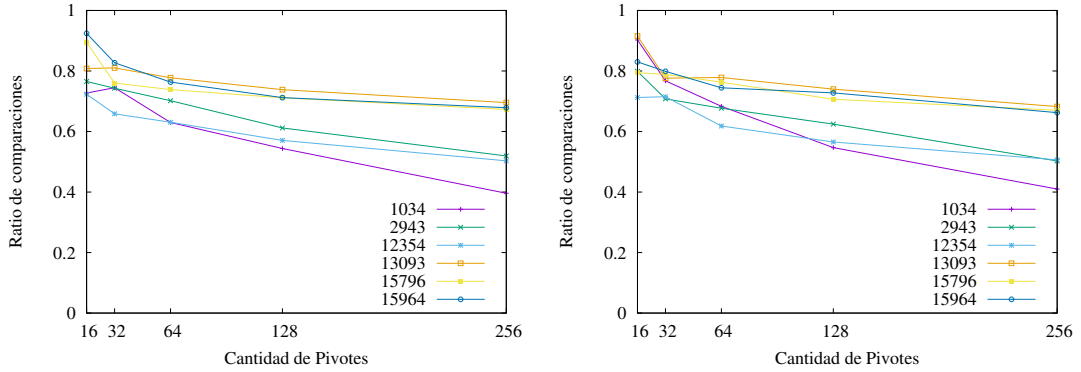
GRAFICA: Efecto de las técnicas de selección de pivotes random vs incremental respecto de evaluaciones de distancia

conclusiones: Como se puede observar, solo en el grupo 1; la técnica de selección incremental realiza menos cantidad de evaluaciones de distancia. En los grupos 2, 3 y 4 es mejor la técnica de selección random solo por muy poca diferencia.

Dado que el comportamiento para ambas técnicas de selección no era el que esperabamos, es decir, incremental no era mucho mejor que random en cuanto a ratio de comparaciones, hicimos un análisis empírico sobre diferentes categorías para entender tal comportamiento.

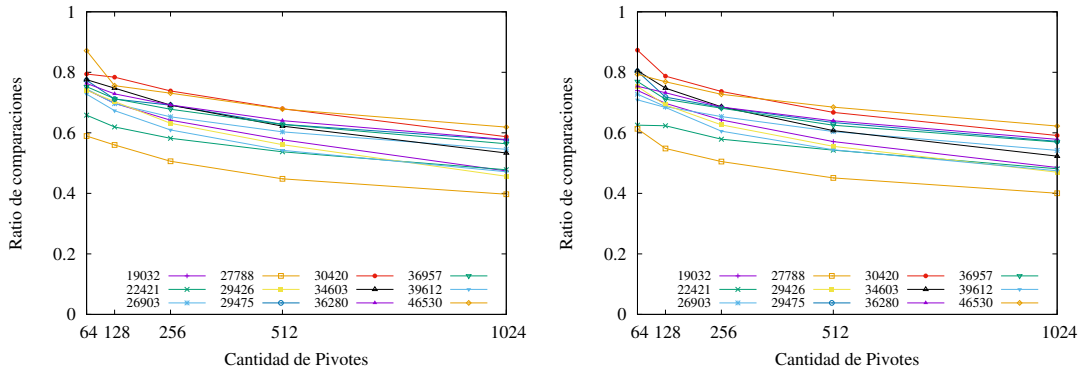
5.3.2. Efecto de la cantidad de pivotes

Las siguientes figuras muestran el efecto de la cantidad de pivotes sobre el ratio de comparaciones por cada base de datos para los distintos grupos. Cada una de las líneas es una base de datos, y estas bases de datos están representadas por su tamaño como se muestra en las gráficas.



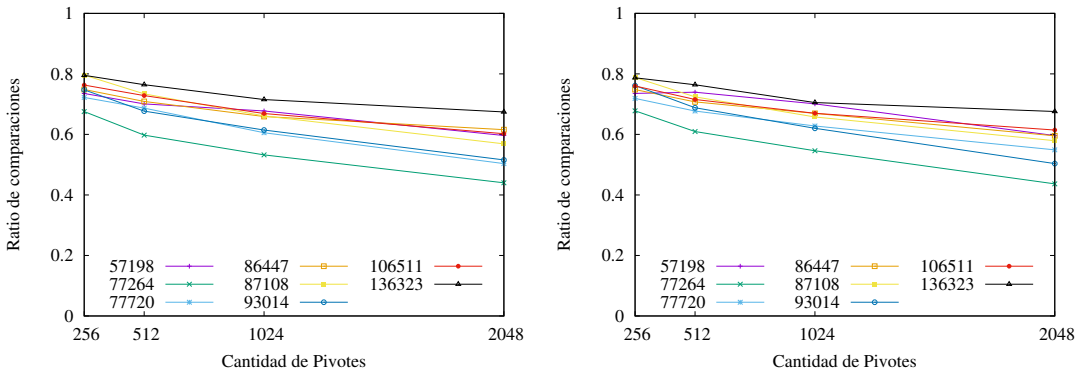
(a) Técnica de selección de pivotes random (b) Técnica de selección de pivotes incremental

Figura 5.3: Grupo 1: Efecto de la cantidad de pivotes sobre el ratio de comparaciones.



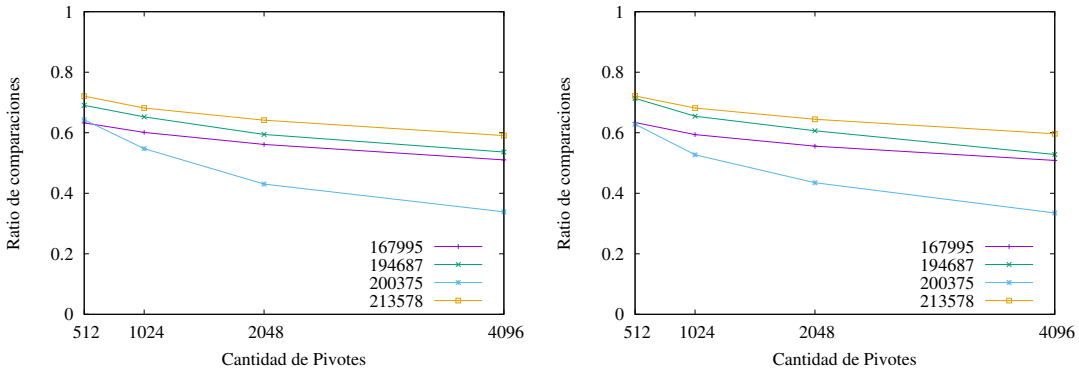
(a) Técnica de selección de pivotes random (b) Técnica de selección de pivotes incremental

Figura 5.4: Grupo 2: Efecto de la cantidad de pivotes sobre el ratio de comparaciones.



(a) Técnica de selección de pivotes random (b) Técnica de selección de pivotes incremental

Figura 5.5: Grupo 3: Efecto de la cantidad de pivotes sobre el ratio de comparaciones.



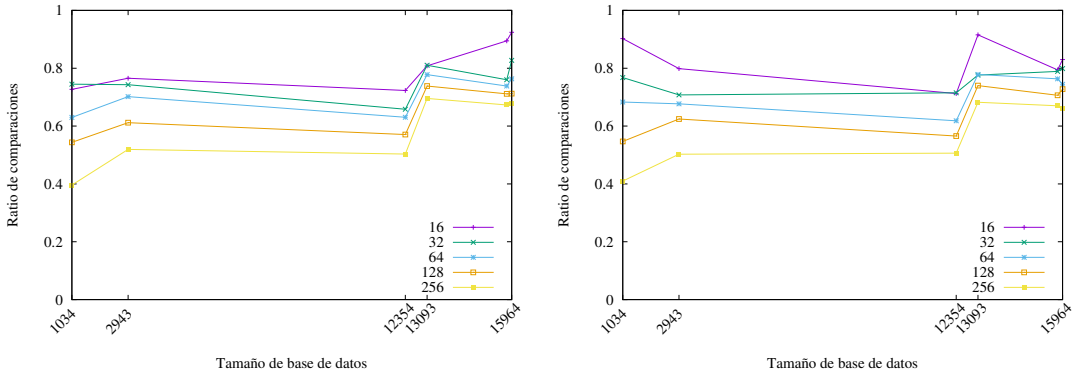
(a) Técnica de selección de pivotes random (b) Técnica de selección de pivotes incremental

Figura 5.6: Grupo 4: Efecto de la cantidad de pivotes sobre el ratio de comparaciones.

Como conclusion se puede observar que para todas las bases de datos de los diferentes grupos a medida que aumenta el número de pivotes, el ratio de comparaciones es menor.

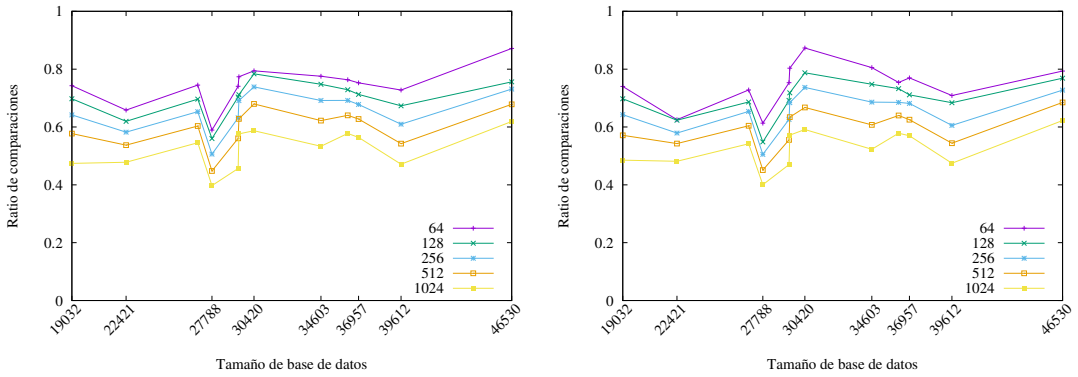
5.3.3. Efecto tamaño de la base de datos

En las siguientes figuras mostraremos el efecto del tamaño de la base de datos sobre el ratio de comparaciones por pivotes.



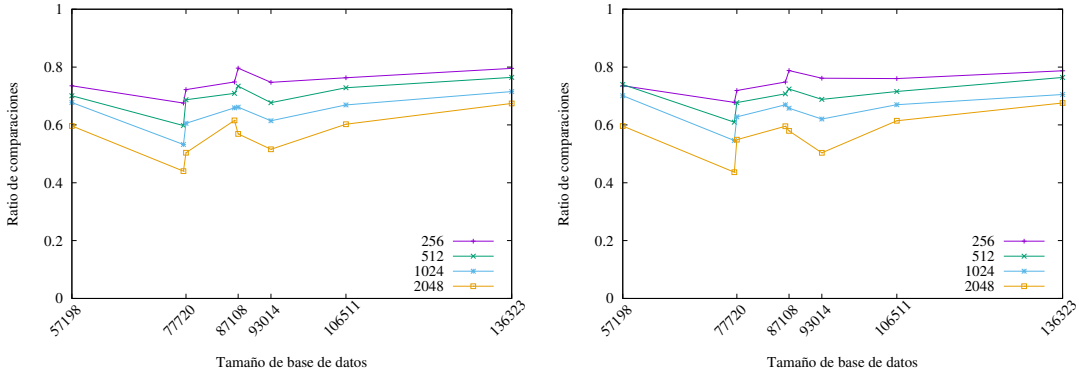
(a) Técnica de selección de pivotes random (b) Técnica de selección de pivotes incremental

Figura 5.7: Grupo 1: Efecto del tamaño de la base de datos sobre el ratio de comparaciones por pivotes.



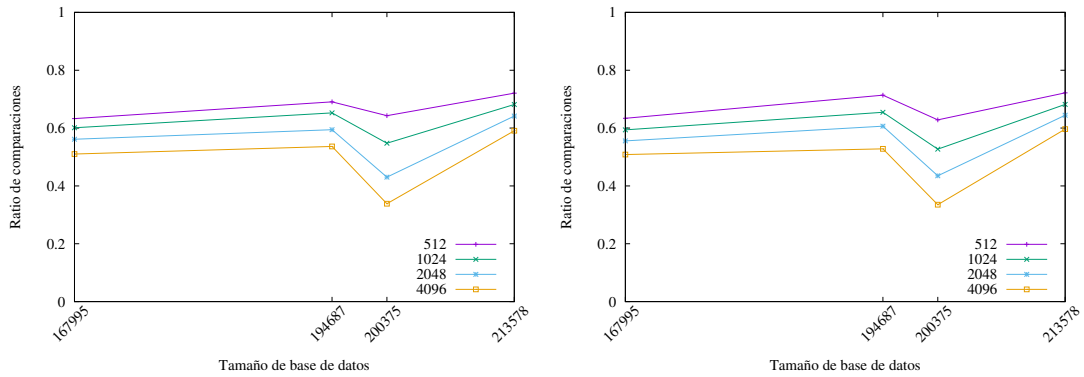
(a) Técnica de selección de pivotes random (b) Técnica de selección de pivotes incremental

Figura 5.8: Grupo 2: Efecto del tamaño de la base de datos sobre el ratio de comparaciones por pivotes.



(a) Técnica de selección de pivotes random (b) Técnica de selección de pivotes incremental

Figura 5.9: Grupo 3: Efecto del tamaño de la base de datos sobre el ratio de comparaciones por pivotes.



(a) Técnica de selección de pivotes random (b) Técnica de selección de pivotes incremental

Figura 5.10: Grupo 4: Efecto del tamaño de la base de datos sobre el ratio de comparaciones por pivotes.

Como conclusión se observa que a medida que aumenta el tamaño de la base de datos, disminuye el porcentaje de filtrado para los distintos grupos de pivotes, en otras palabras aumenta el porcentaje de comparaciones respecto del tamaño de la base de datos.

Capítulo 6

Conclusiones

6.1. Contribuciones

Índice de figuras

Índice de cuadros

Bibliografía

[AD09] Anthony D., Smith S., W. T. *Reputation and reliability in collective goods: The case of the online encyclopedia wikipedia.* Rationality & Society 21, 2009.

[AFE10] Anthony Fader, S. S., and Etzioni, O. *Identifying Relations for Open Information Extraction.* 2010.

[AM04] Angeles, P., and MacKinnon, L. *Detection and Resolution of Data Inconsistences, and Data Integration using Data Quality Criteria.* QUATIC'2004, 2004.

[Ami03] Amirijoo, M. *Specification and Management of QoS in Imprecise Real-Time Databases.* Proceeding of the Seventh International Database Engineering and Applications Symposium, 2003.

[And13] Anderka, M. *Analyzing and Predicting Quality Flaws in User-generated Content: The Case of Wikipedia.* 2013.

[AS10] Anderka, M., and Stein, B. *A Breakdown of Quality Flaws in Wikipedia.* 2010.

[Blu08] Blumenstock, J. E. *Size matters: word count as a measure of quality on Wikipedia.* ACM. ISBN 978-1-60558-085-2. doi: 10.1145/1367497.1367673, 2008.

[BM07] Banko M., Cafarella M. J., S. S. B. S. y. E. O. *Open information extraction from the Web.* In the Proceedings of the 20th International Joint Conference on Artificial Intelligence, 2007.

[Bou01] Bouzeghoub, M. y Kedad, Z. *Quality in Data Warehousing. Information and Database Quality.* M. Piattini, C. Calero y M. Genero, Kluwer Academic Publishers, 2001.

[BP04] Bouzeghoub, M., and Peralta, V. *A Framework for Analysis of data Freshness.* Proc. IQIS2004, 2004.

[BSG05] Besiki Stvilia, Michael B. Twidale, L. C. S., and Gasser, L. *Assessing information quality of a community-based encyclopedia.* MIT, 2005.

[BSG08] B. Stvilia, M. Twidale, L. C. S., and Gasser., L. *Information quality work organization in wikipedia.* 2008.

[Cap02] Cappiello, C. *A Model of Data Currency in Multi-Channel Financial Architectures.* 2002.

[Cap04] Cappiello, C. *Data quality assessment from the user's perspective.* Proc. IQIS2004, 2004.

[Cra01a] Crawford, H. *Encyclopedias.* 3ra Edición. Englewood, CO: Libraries Unlimited., 2001.

[Cra01b] Crawford., H. *Encyclopedias. In Richard E. Bopp and Linda C. Smith, editors, Reference and information services: an introduction.* 2001.

[DB03] Davis, G.F., Y. M., and Baker, W. *The small world of the American corporate elite 1982-2001.* Strategic Organization, Vol. 1 No. 3, 2003.

[Epp02] Eppler, M. y Muenzenmayer, P. *Measuring Information Quality in the Web Context: A Survey of State-of-the-Art Instruments and an Aplication Methodology.* 2002.

[Etz12] Etzioni, O., B. M. S. S. W. D. *Open information extraction from the web.* 2012.

[Fel98] Fellbaum, C., e. *WordNet: an electronic lexical database.* MIT Press, 1998.

[Fin] Finkelstein, C. y Aiken, P. *XML and Corporate Portals.* http://www.wilshireconferences.com/xml/paper/xml-portals.htm.

[Fug02] Fugini, M. *Data Quality in Cooperative Web Information Systems.* 2002.

[FW10] Fei Wu, D. S. W. *Open Information Extraction using Wikipedia.* 2010.

[Gas01] Gasser, L., . S. B. *A new framework for information quality.* Champaign: University of Illinois at Urbana-Champaign., 2001.

[Ger04] Gertz, M. *Report on the Dagstuhl Seminar "Data Quality on the Web".* SIGMOD Record, vol. 33, Nº 1, 2004.

[Hai03] Haider, A. y Koronios, A. *Authenticity of Information in Cyberspace: IQ in the Internet, Web, and e-Business.* 2003.

[JG99] Juran, J. M., and Godfrey, A. B. *Juran's quality handbook.* McGraw-Hill London., 1999.

[Joh98] Johnson, R., W. D. *Applied multivariate statistical analysis (4th ed.).* Upper Saddle River, NJ: Prentice-Hall., 1998.

[Juf09] Juffinger, A., G. M. L. E. *Blog credibility ranking by exploiting verified content.* In: Proceedings of the Workshop on Information Credibility on the Web (WICOW) in conjunction with WWW'2009, New York, NY, USA, ACM (2009) 51-58, 2009.

[Jur92] Juran, J. *Juran on quality by design.* 1992.

[Kat99] Katerattanakul, P. y Siau, K. *Measuring Information Quality of Web Sites: Development of an Instrument.* Proceeding of the 20th International Conference on Information System., 1999.

[KJ93] Kim J., M. D. *Acquisition of semantic patterns for information extraction from corpora.* Proceedings of Ninth IEEE Conference on Artificial Intelligence for Applications, 1993.

[KS02] Kakihara, M., and Sørensen, C. *Exploring knowledge emergence: from chaos to organizational knowledge.* Journal of Global Information Technology Management, Vol. 5 No. 3, 2002.

[Lee02] Lee, Y. *AIMQ: a methodology for information quality assessment. Information and Management.* Elsevier Science, 2002.

[Lex12] Lex, E., V. M. E. M. F. E. C. L. H. C. S. B. G. M. *Measuring the quality of web content using factual information.* In 2nd joint WICOW/AIRWeb workshop on Web quality (WebQuality'12). ACM., 2012.

[LGD09] Loris Gaio, Matthijs den Besten, A. R., and Dalle., J.-M. *Wikibugs: using template messages in open content collections.* ACM. ISBN 978-1-60558-730-1. doi: 10.1145/1641309.1641330., 2009.

[Lih04] Lih, A. *Wikipedia as Participatory Journalism: Reliable Sources? Metrics for evaluating collaborative media as a news resource.* Journalism and Media Studies Centre University of Hong Kong, 2004.

[Lim00] Lim, O. y Chin, K. *An Investigaction of Factors Associated With Customer Satisfaction In Australian Internet Bookshop Websites.* 3rd Western Australian Workshop on Information Systems Research, Autralia., 2000.

[LJ11] Liu J., R. S. *Who does what: Collaboration patterns in the wikipedia and their impact on article quality.* ACM Transactions on Management Information Systems 2, 2011.

[LP13] Lian Pohn, Edgardo Ferretti, M. E. *Evaluación de la Calidad de la Información de Wikipedia en Español.* http://sedici.unlp.edu.ar/bitstream/handle/10915/27332 /Documento_completo.pdf?sequence=1, 2013.

[LS10] Lipka, N., and Stein, B. *Identifying featured articles in Wikipedia:writing style matters.* ACM. ISBN 978-1-60558-799-8. doi: 10.1145/1772690.1772847., 2010.

[MAB12] M. Anderka, B. S., and Busse, M. *On the Evolution of Quality Flaws and the Effectiveness of Cleanup Tags in the English Wikipedia.* 2012.

[MAC05] M. Angélica Caro, Coral Calero, I. C. M. P. *Data Quality in Web Applications: A State of the Art.* 2005.

[Mag10] Magdy, A., W. N. *Web-based statistical fact checking of textual documents.* In: Proceedings of the 2nd international workshop on Search and mining user-generated contents. SMUC '10, New York, NY, USA, ACM (2010) 103-110, 2010.

[MAL11] Maik Anderka, B. S., and Lipka., N. *Towards automatic quality assurance in Wikipedia.* ACM. ISBN 978-1-4503-0637-9. doi: 10.1145/1963192.1963196., 2011.

[MAL12] M. Anderka, B. S., and Lipka, N. *Predicting Quality Flaws in Usergenerated Content: The Case of Wikipedia.* In 35th Annual SIGIR Conference. ACM, 2012.

[Mar03] Marchetti, C. *Enabling Data Quality Notification in Cooperative Information Systems through a Web-service based Architecture.* 2003.

[MBD11] Michael Bendersky, W. B. C., and Diao., Y. *Quality-biased ranking of web documents.* ACM. ISBN 978-1-4503-0493-1. doi: 10.1145/1935826.1935849., 2011.

[MI12] Myshkin Ingawale, Amitava Dutta, R. R. P. S. *Network analysis of user generated content quality in Wikipedia.* 2012.

[Mol] Molina, M. P. *Calidad y evaluacion de los contenidos electrónicos.* http://www.mariapinto.es/e-coms/eva_con_elec.htm.

[OFR12] Oliver Ferschke, I. G., and Rittberger, M. *FlawFinder: A Modular System for Predicting Quality Flaws in Wikipedia.* 2012.

[Per02] Pernici, B. y Scannapieco, M. *Data Quality in Web Information Systems.* Proceeding of the 21st International Conference on Conceptual Modeling, 2002.

[PT10] Pirkola, A., and Talvensaari., T. *A topic-specific web search system focusing on quality pages.* Springer. ISBN 3-642-15463-8. doi: 10.1007/978-3-642-15464-5_64., 2010.

[R.97] R., H. *Evaluating Internet Research Sources.* http://www.sccu.edu/faculty/R_Harris/evalu8it.htm, 1997.

[SEMZ09] Stuart E. Madnick, Richard Y. Wang, Y. W. L., and Zhu., H. *Overview and framework for data and information quality research.* Journal of data and information quality, 1(1):1–22, 2009. ISSN 1936-1955. doi:10.1145/1515693.1516680., 2009.

[Sta00] Stacey, R. *The emergence of knowledge in organizations", Emergence: Complexity and Organization.* Vol. 2 No. 4, 2000.

[Str97] Strong, D. *Data Quality in Context.* Communications of the ACM. Vol. 40. Nº 5., 1997.

[Suc07] Suchanek, F.M., K. G. W. G. *Yago: A core of semantic knowledge.* In: Proceedings of the 16th International Conference on World Wide Web. WWW '07, New York, NY, USA, ACM (2007) 697-706, 2007.

[US05] Uzzi, B., and Spiro, J. *Collaboration and creativity: the small world problem.* American Journal of Sociology, Vol. 111 No. 2, 2005.

[WD07] Wilkinson D., H. B. *Cooperation and quality in Wikipedia.* In 3rd international symposium on wikis and open collaboration, 2007.

[WF94] Wasserman, S., and Faust, K. *Social Network Analysis: Methods and Applications.* Cambridge University Press, New York, NY., 1994.

[Wik] Wikipedia. *http://en.wikipedia.org/wiki/Wikipedia: Verifiability,_not_truth.*

[WS96] Wang, R. Y., and Strong, D. M. *Beyond accuracy: what data quality means to data consumers.* Journal of management information systems, 12(4):5–33, 1996. ISSN 0742-1222., 1996.

[ZC05] Zhou, Y., and Croft., W. B. *Document quality models for Web ad hoc retrieval.* ACM. ISBN 1-59593-140-6. doi: 10.1145/1099554.1099652., 2005.

[ZG00] Zhu, X., and Gauch., S. *Incorporating quality metrics in centralized/distributed information retrieval on the World Wide Web.* ACM. ISBN 1-58113-226-3. doi: 10.1145/345508.345602., 2000.

[Zhu] Zhu, Y. y Buchmann, A. *Evaluating and Selecting Web Sources as external Information Resources of a Data Warehouse.*